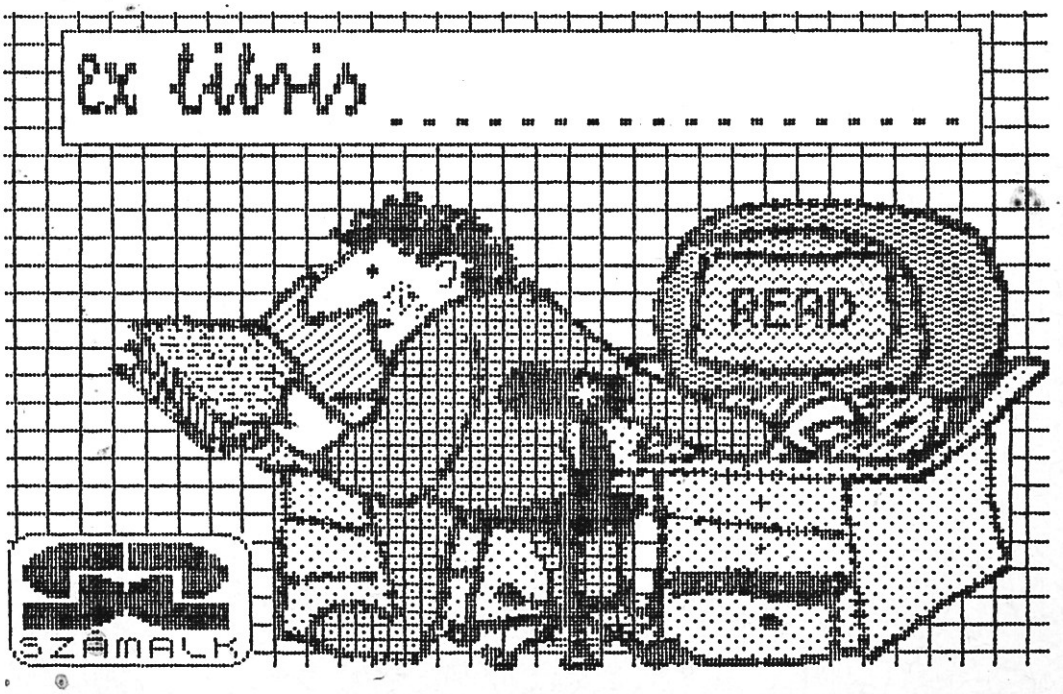
Csépai János - Quittner Pál

BIT-LES

számítástechnikai lemezlovas

SZÁMALK

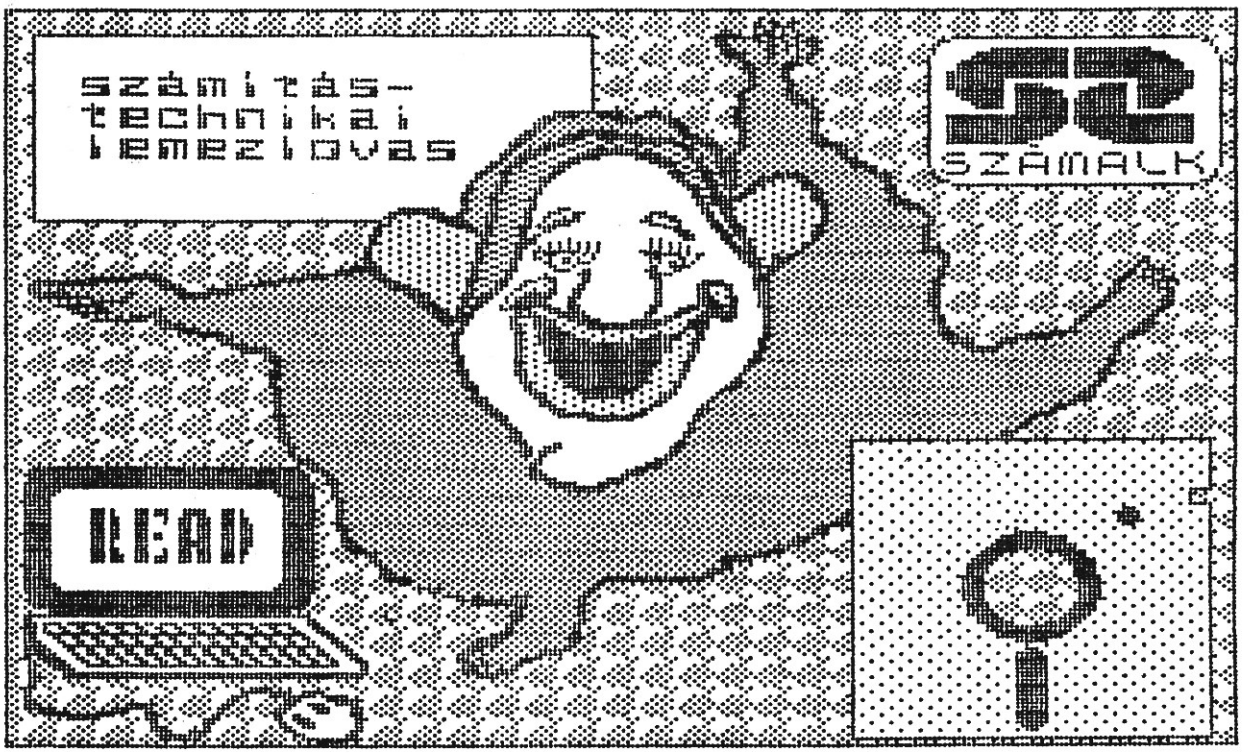

Ex Libris
READ
SZÁMALK

Miklósnak, Balázsnak
és
Katinak,
de leginkább Zsoltnak,
családunk számítástechnikai
szakembereinek,
és könyvünk legszigorúbb lektorainak.

A két apa és nagyapó

Csépai János - Quittner Pál:

BIT-LES



SZÁMALK

1987

A kötetet Brückner Huba és Kis Ádám szerkesztette.
Lektorálta: Varga András

A karikatúrákat Halász Géza készítette.

© Csépai János és Quittner Pál, 1987.
© Halász Géza, 1987.

ISBN 963 553 126 5

Kiadja a Számítástechnika-alkalmazási Vállalat
Felelős kiadó: Havass Miklós, vezérigazgató
Műszaki vezető: Nagy Sándor
A fedelet Szücs Édua tervezte
Tipográfia: Beleznai László
Megjelent Fr/4 formában, 23 (A/5) ív terjedelemben
Készült a Kaposvári Nyomdában
A nyomdai megrendelés törzsszáma: 171704

TARTALOMJEGYZÉK

A SZERZŐK MENTEGETŐDZÉSE (Előszó) 7

MIELŐTT ELKEZDENÉNK

1. BEVEZETÉS HELYETT 11
2. MI KAPHATÓ EGY VASKERESKEDÉSBEN? (A hardver és a szoftver) 13
3. A MIÉNK MÁR EGY MÁSIK GENERÁCIÓ (Számítógép-generációk) 16
4. MANCIKA MINT SZÁMÍTÓGÉP (Az adatfeldolgozás főbb folyamatai) 20
5. AZ ÉN GÉPEM, AZ ÉN IGEN KIS GÉPEM (Számítógépek osztályozása ár és teljesítmény szerint) 22
6. ÁGYÚVAL VERÉBRE? (Milyen gépet válasszunk magunknak?) 26
7. KI MICSODA? (Számítástechnikai munkakörök) 29
8. SZÁMÍTÁSTECHNIKA GÖRBE TÜKÖRBŐL – AZAZ BELÜLRŐL NÉZVE (Murphy (sz)ámítástechnikai törvényei) 33

VÁGJUNK BELE! A HARDVER

9. A SZÁMÍTÓGÉPEK MŰKÖDÉSI ELVE 37
10. HOL TALÁLLAK? (Bit, bájt, tárolócímzés) 39
11. A BUTA SZÁMÍTÓGÉP, AMELYIK KETTŐIG SEM SZÁMOL 41
12. A HARDVER LEGFONTOSABB ELEMEI 43
13. CSAK A SZÉPRE EMLÉKEZEM . . . (ROM és RAM) 47
14. MI A TEENDŐ? (Utasítások) 49
15. KI KERÜLHET AZ FTC ELÉ? (Betűk, számok, jelek ábrázolása) 52
16. NEMCSAK KETTESBEN SZÉP AZ ÉLET (Számábrázolás) 54
17. A TÉVEDHETETLENSÉG TITKA (Hardverhibák elleni védelem) 56
18. (NE) JÁTSSZUNK A KÁRTYÁKKAL! (A személyi számítógép architektúrája 58
19. PÁRBESZÉD A GÉPPEL (Adatmegjelenítők) 60
20. AZ ÍRÁS MEGMARAD (Érintéses nyomtatók) 62
21. NE TAPOGASS! (Nem érintéses és színes nyomtatók) 65
22. LEMEZTELENÍTVE NEM MEGY! (Mágneslemezek) 67
23. FORMÁS LEMEZEK – LEMEZFORMÁK (A személyi számítógép háttértárai) 70
24. MIT LÁTNAK SZEMEIM! (Optikai jelolvasók) 73
25. VONALASAK LESZÜNK!(?) (Vonalkódok) 75
26. LEXIKON EGY OLDALON (Optikai tárak) 77
27. BE VAGYUNK HÁLÓZVA (Távfeldolgozó hálózatok) 79
28. TÚL AZ ÓPERENCIÁN! (Adatátviteli és számítógép-hálózatok) 82
29. SZÁMÍTÁSTECHNIKA A HARDVERES GÖRBE SZEMÉVEL 84

AMITŐL INTELLIGENS: A SZOFTVER

30. GÉP TESTBEN GÉPEN ÉLEK (A szoftver legfontosabb összetevői) 89
31. HOGYAN LESZ A BOLHÁBÓL ELEFÁNT? (Operációs rendszerek) 92

32. HÉT NYELVEN BESZÉL (Programozási nyelvek) 95
33. KÖZÖS GÉPNEK NEM TÚROS A HÁTA (Multiprogramozás) 97
34. PROTEKCIÓSOK ELŐRE! (A multiprogramozás ütemezése) 100
35. MINDENKINEK SZÜKSÉGLETE SZERINT (Memóriagazdálkodás) 102
36. CSAK AKKOR MAGAD, HA SZOLGÁD NINCSEN (Szolgáltatóprogramok) . . 104
37. A SZOFTVER ÉRTÉKE 106
38. KALÓZKODÁS A TENGERNYI SZOFTVERBEN (A szoftver jogi és fizikai védelme) 109
39. VIGYÁZAT, ADATTOLVAJ! (Számítógépes rendszerek biztonsága) 113
40. ADATVÉDELEM, ADATBIZTONSÁG 115

DOLGOZ(Z)UNK A SZÁMÍTÓGÉPPEL!

41. AKINEK AZ ISTEN GÉPET ADOTT. . . . (Miért nehéz programot írni?) 121
42. MÓDSZER ÉS CÉL (A problémamegközelítés problémája) 123
43. A REKORDOK ÁLLOMÁNYA (File, rekord, adatelem) 125
44. ITT A PIROS – HOL A PIROS? (Adatszervezési alapfogalmak) 127
45. HA SZEMÉT MEGY BE, SZEMÉT IS JÖN KI (Adatrögzítés, adatellenőrzés) . 131
46. KÉTSZER KETTŐ NÉHA ÖT (Műveletek kerekített számokkal) 134
47. ÍGY PROGRAMOZUNK MI (?) (Milyen a jó program és a jó programozó) . . . 136
48. SZABADÍTS MEG A GONOSZTÓL! (Hibakeresés számítógépes programokban) 139
49. BOCSÁSD MEG A VÉTKEINKET! (Gyakori programozási hibák) 143
50. TÍZ PARANCS (Operációs rendszerek gyakori parancsai) 145
51. ÍGY PROGRAMOZTOK TI (ÉS ŐK IS) (Számítástechnika alulnézetben, a leggörbébb szoftveres szemmel nézve) 148
52. AZ AMATŐR PROFI ÉS A PROFI AMATŐR (Szakma-e a számítástechnika?) . 150
53. BELÉPÉS CSAK KÜLÖN ENGEDÉLLYEL (Látogatás egy számítóközpontban) 154

VAJON MIRE JÓ AZ EGÉSZ?

54. PONT–PONT–VESSZŐCSKE (Számítógépes grafika) 159
55. OLLÓ ÉS RAGASZTÓ NÉLKÜL IS MEGY (Szövegszerkesztők) 166
56. ÉKTELENÜL ÉKES (A magyar ábécé problémái) 169
57. MINDENT EGY HELYEN (Adatbázisok, adatbankok) 171
58. SZÉP IS LEGYEN, JÓ IS LEGYEN! (Dokumentum-visszakereső rendszerek) . 175
59. A BÜROKRATA RÉMÁLMA (Az automatizált iroda) 178
60. DR. (MED) UNIV. SZÁMÍTÓGÉP (Számítógépes diagnosztizálás) 181
61. SZIMULÁL(J)UNK! (Kiképzés számítógép segítségével) 183
62. MIKOR ESIK LE A TANTUSZ? (Elektronikus pénz) 185
63. TUVUDSZ IVÍGY BEVESZÉVÉLNIVI? (Rejtjelezés számítógéppel) 187
64. A PÉNZ BESZÉL – A GÉP DOLGOZIK (A számítógépes feldolgozások költségei) 189
65. ELŐTTÜNK AZ ÖZÖNVÍZ (Elemi csapások, katasztrófák hatásának elhárítása) 192
66. VERSENYBEN AZ EMBERI AGGYAL (Szakértői rendszerek) 195
67. ÁLDÁS ÉS ÁTOK (A növekedés határai) 198
68. NAGY NEVEK, KIS TÖRTÉNELEM 201
69. EPILÓGUS HELYETT (Búcsú az olvasótól) 213

SZÁMÍTÁSTECHNIKAI HOL–MI (Tárgymutató) és MI–MI (Szótárka) 217

A SZERZŐK MENTEGETŐDZÉSE
(Előszó)

A disc jockey (magyarul lemezlovas) két zeneszám között érdekes történetekkel (magyarul pletykákkal) szórakoztatja közönségét. Ily módon hasznosan múlik el a lemezcserék alatti szünet.

A szerzők, felvállalva a floppy disk jockey — magyarul hajlékonylemez-lovas — cseppet sem irígylésre méltó szerepét, szeretnék a személyiszámítógép-használók széles tömegét megnyerni, hogy üres perceikben (amikor úgyis csak a gép csinál valamit) hasznos ismeretek szerzése céljából lapozzák fel e könyvet egy-egy csipetnyi (vagy chip-etnyi?) számítástechnika erejéig.

Nem akarunk 70-kedni, éppen ezért mondanivalónkat nem hetven, hanem 69 falatban (a könyvkiadás hagyományai szerint: fejezetben) gyűjtöttük össze. Tudjuk, hogy a géppel foglalkozó kezdő és haladó amatőrök borzasztóan elfoglalt emberek, így nem is reménykedünk abban, hogy bárki is elejétől végig egyszuszra elolvassa valamennyi fejezetet. Ezért minden fejezetben (mintegy enciklopédikus jelleggel) más-más problémakört szeretnénk megvilágítani. Ennek következménye, hogy kisebb-nagyobb átfedések előfordulhatnak. Legyen mentségünkre, hogy az életben is összefüggenek a dolgok, továbbá, hogy az ismert mondás szerint ismétlés a tudás anyja.

S ha valaki nem szeret ismételni, viszont elfelejt dolgokat, még nem kell szidnia a tudás anyját. Rajtuk is segíteni kívánunk, akik régi jó diákszokás szerint „tudták, de elfelejtették" a fogalmakat. Ha legalább azt tudják, hogy mi az amit elfelejtettek, akkor már nincs semmi baj. Csak fel kell lapozni a könyv végén levő tárgymutatót (amely már létével is „tudományosabbá" teszi munkánkat és „megbocsáthatóbbá" a szórakoztató rajzokat). Ha kis szerencséjük van, akkor a tárgymutató segítségével megtalálják az ebben az esetben „minilexikonként" is használható könyvben a megfelelő, rövid ismertetést.

Persze az is előfordulhat, hogy ezzel az elgondolással tévedünk. Mentségünkre legyen, hogy

TÉVEDNI EMBERI DOLOG.

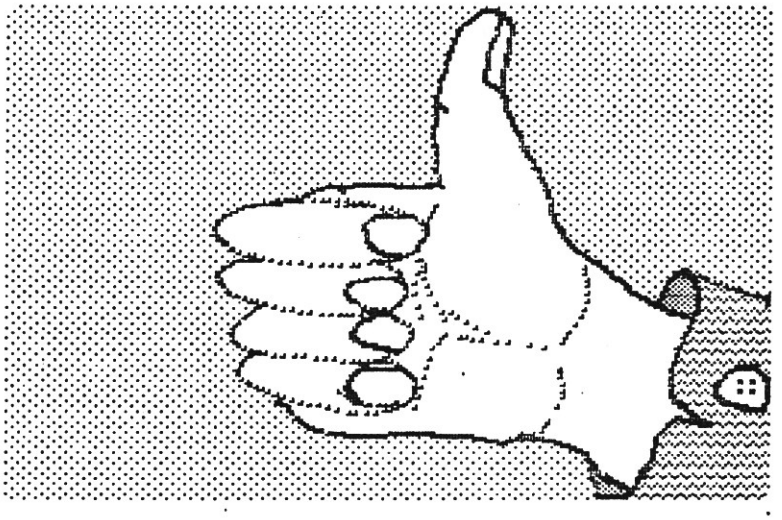
De ne felejtse el a kedves olvasó, hogy

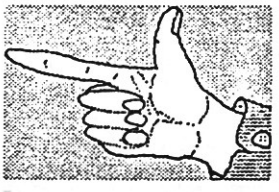
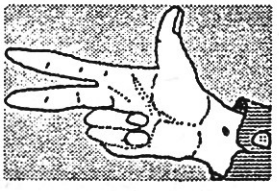
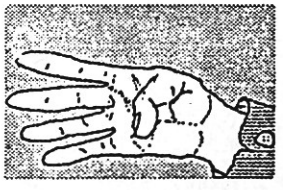
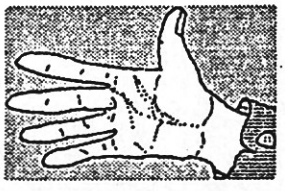
A DOLGOKAT IGAZÁN ÖSSZEKUTYULNI
CSAK SZÁMÍTÓGÉPPEL LEHET.

(Nem) ehhez kívánunk sikereket mindazoknak, akik egy chipetnyi vagy bitnyi ismeretért fellapozzák könyvünket.

Budapest, 1987. május

MIELŐTT ELKEZDENÉNK

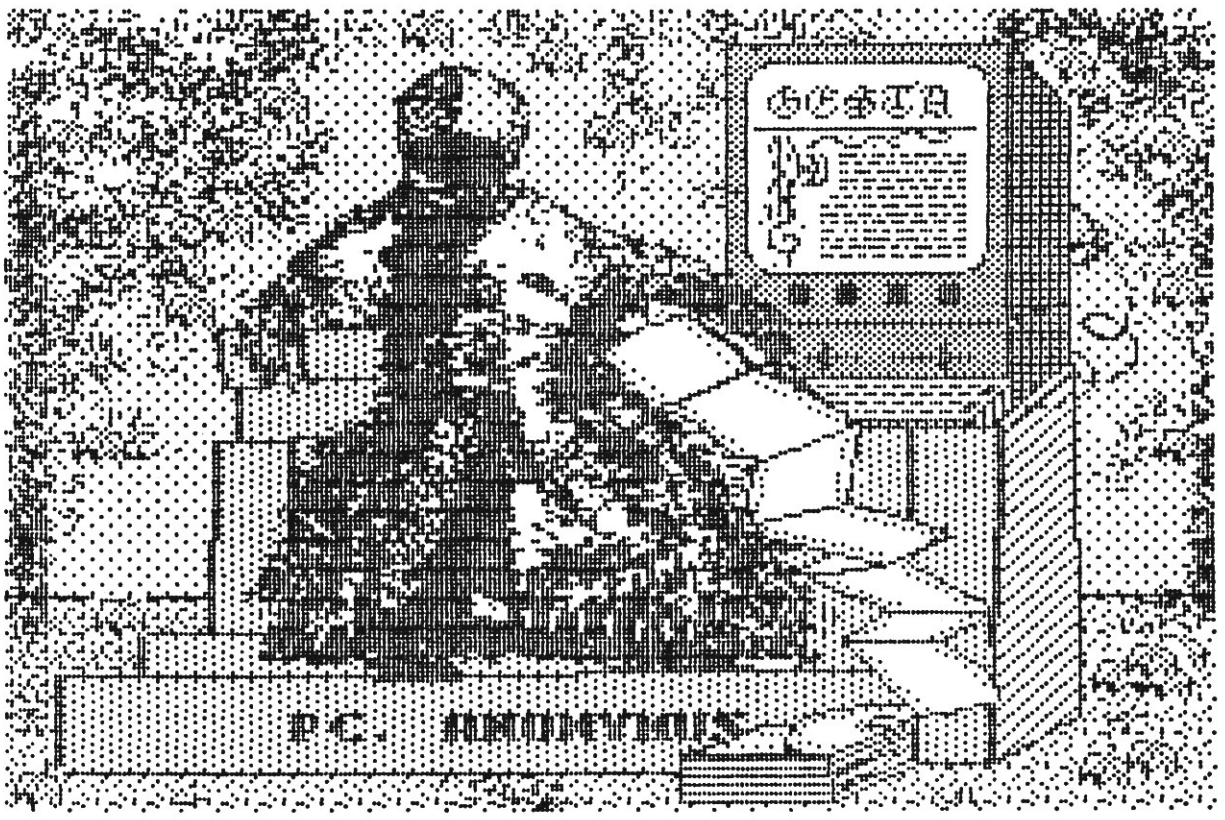


1.

BEVEZETÉS HELYETT

Az elmúlt évtized technikai fejlődésének egyik legjellemzőbb eredménye, hogy a mikroelektronika, a számítógépek bevonultak mindennapi életünkbe. A régi feudális uralkodó helyett (aki úgy képzelte el népe jólétét, hogy minden alattvalójának tyúk főjön a fazekában), a mai kor miniszterelnökének az lehetne a kívánsága, hogy minden , – legalábbis reá szavazó – állampolgárnak számítógép legyen az asztalán. Az iparilag erősen fejlett országokban húsz, nálunk tíz évvel ezelőtt még gyakran előfordult, hogy a számítógépet inkább státuszszimbólumként és nem hasznos, gazdaságilag indokolt munkaeszközként kezelték egyes vállalatoknál, intézményeknél. Ma pedig már nem csak ezeken a helyeken vált nélkülözhetetlenné, hanem sok ember mindennapi életében is. Tizenéves gyerekek az iskolákban és szakfoglalkozásokon már Magyarországon is olyan természetességgel dolgoznak a mikroszámítógépekkel, mintha nem is lovas, hanem számítógépes nemzet leszármazottjai lennének.



...mintha nem is lovas, hanem számítógépes nemzet fiai lennénk...

Az ipari forradalom a gépek megjelenésével megszabadította az embert a nehéz testi munkától, és megsokszorozta fizikai teljesítőképességét. A szemünk láttára kibontakozó „második ipari forradalom", a számítógépek felhasználásával megvalósuló „információs forradalom" ugyanezt teszi a szellemi munka területén.

Az ember szellemi munkájának megkönnyítése és gyorsabbá tétele szempontjából a legutolsó négy évtized többet jelentett, mint az addig eltelt többezer éves időszak. A korábbi számítási segédeszközök, kezdve az abakusztól (golyós összeadó) és a logarléctől az asztali

számológépekig, ideértve még az első elektronikus számítógépeket is, csupán a számítások meggyorsítására szolgáltak. A modern elektronikus számítógépek viszont technikai lehetőségeik révén helyet kaptak az élet egyéb területein: a gazdaságirányításban, az üzleti ügyvitelben, a műszaki folyamatok irányításában és még sok egyéb helyen is. A mikroelektronikai alkatrészek és a kapcsolódó berendezések árának rohamos csökkenése pedig nemcsak lehetővé, hanem gazdaságossá is tette ezeknek a gépeknek az alkalmazását. Még akkor is, hogyha nálunk bizonyos gépek elterjedését egyéb okokból indokolatlanul magas „védővámokkal" kívánták megakadályozni, vagy legalábbis megnehezíteni. (A szerzők remélik, hogy bármilyen gyors lesz a könyv nyomdai átfutása, megjelenésekor már jogos lesz a múlt idő használata a vámokkal kapcsolatban.)

A továbbiakban megkíséreljük a számítástechnika legfontosabb fogalmait, eszközeit röviden, közérthetően megmagyarázni azoknak, akiket még nem kapott el a „komputeritisz" nevű betegség, de alkatilag már alkalmasak a megfertőződésre. Ebből a szempontból a legérzékenyebb a tizenéves korosztály, mely jelen esetben reméljük 12 évtől 99 éves korig tart, hiszen nincsenek öreg emberek, csak elavult témák. A számítástechnika és alkalmazása azonban minden bizonnyal nem tartozik ezek közé.

Természetesen nem akarjuk misztifikálni a számítástudományt, nem akarjuk azt elhitetni az olvasóval, hogy a számítógépek a legokosabbak, azokkal lehet minden feladatot a leggyorsabban megoldani. Erre vonatkozólag álljon itt egy megtörtént eset. A manapság oly divatos számítástechnikai tanfolyamok egyikén a hallgatók némi programozási ismeret elsajátítása után rendszeresen megkapták azt a feladatot, hogy keressék meg azokat a kétjegyű számokat, melyeknek négyzete „aabb" alakú. A jó logikával rendelkező tanulók 15–20 perc alatt készítették el a hibátlan programot. Az egyik alkalommal egy fiatalember két perc múlva jött a válasszal: 88 az egyetlen ilyen szám, és ennek négyzete 7744.
— Ilyen hamar megírta a programot? — kérdeztük csodálkozva.
Nem — válaszolta — csak elővettem a négyjegyű függvénytáblázatomat, abban megkerestem a számok négyzetét, és azonnal megtaláltam az egyetlen helyes megoldást.

Tulajdonképpen máig sem tudtuk eldönteni, hogy örüljünk-e, vagy sajnáljuk, hogy ez az eset nem tipikus. Benne van ugyanis egy nagyon fontos gondolat. Mielőbb egy probléma megoldásához számítógépet használnánk, gondoljuk végig, nem lehet-e azt olcsóbban, megfelelő határidőre egyszerűbb eszközökkel megoldani. Ha nem — és valóban nagyon sokszor ez lesz az elemzés eredménye — akkor, de csak akkor, irány a számítógép! E könyv célja megtanítani az olvasót, hogy miképpen kell korunk egyik legcsodálatosabb technikai vívmányát, a számítógépet a megfelelő helyre tenni. Ezalatt természetesen nem azt értjük, miként tegye ki az íróasztalára, hogy kényelmesen hozzáférhessen a billentyűzethez, és jól lássa a képernyőt. Azt sem, hogyan helyezze el a gép működését leíró szakirodalmat jól látható helyen a könyvespolcra, hadd lássa mindenki, mivel foglalkoznak itt. Azt kívánjuk megmutatni, miből állnak és hogyan is működnek a számítógépek. Mire használhatók ésszerűen, de azt is, hogy mire nem alkalmasak? Miben várhatunk tőlük, és miben adnak segítséget? Legfőképpen pedig azt, hogy mit kell nekünk magunknak tenni annak érdekében, hogy a számítógépeket jól, gazdaságosan, értelmes feladatok megoldására használhassuk.

A számítógép önmagában nem csodaszer, egymagában semmit sem old meg. Ahhoz, hogy valóban hatékonyan alkalmazzuk, szükségünk van megfelelő programokra, munka- és ügyvitelszervezésre, a számítógép precíz munkájához szükséges adatok szolgáltatására. És ha mindez együtt van, még akkor is nélkülözhetetlen az ember, aki a gép és a programok által szolgáltatott eredményeket értékeli, értelmezi, elemzi, és végül ezen tevékenységek alapján cselekszik, dönt.

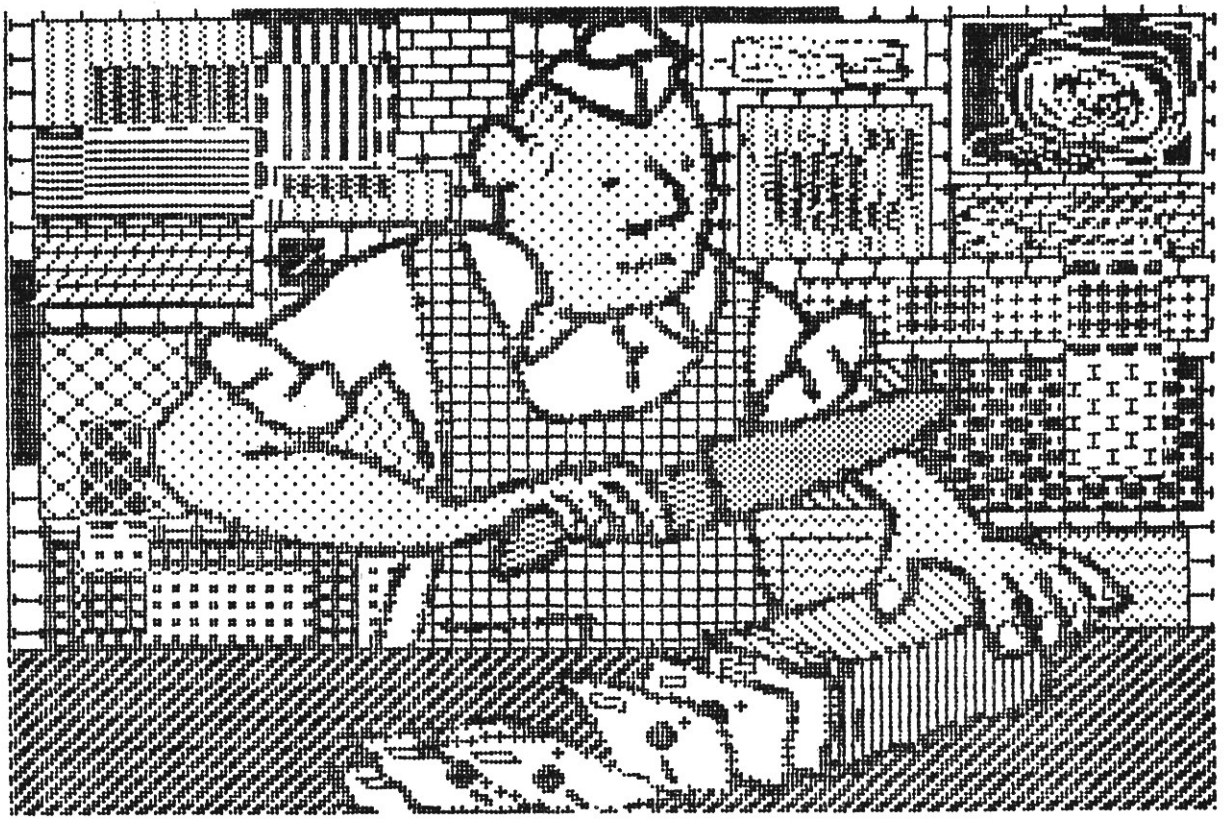
2.

MI KAPHATÓ EGY VASKERESKEDÉSBEN?
(A hardver és a szoftver)

„Az Isten nem bottal ver,
Arra ott van a hardver!"

– így vigasztalják magukat a programozók, amikor a programjuk a tizedik nekifutásra is „meghal" valamilyen általuk teljesen érthetetlen hiba miatt.

De mit is értünk a számítástechnikában ezen a szép, az angol hardware szóból fonetikusan magyarrá varázsolt kifejezésen és a párján, a software-ből származó szoftveren?



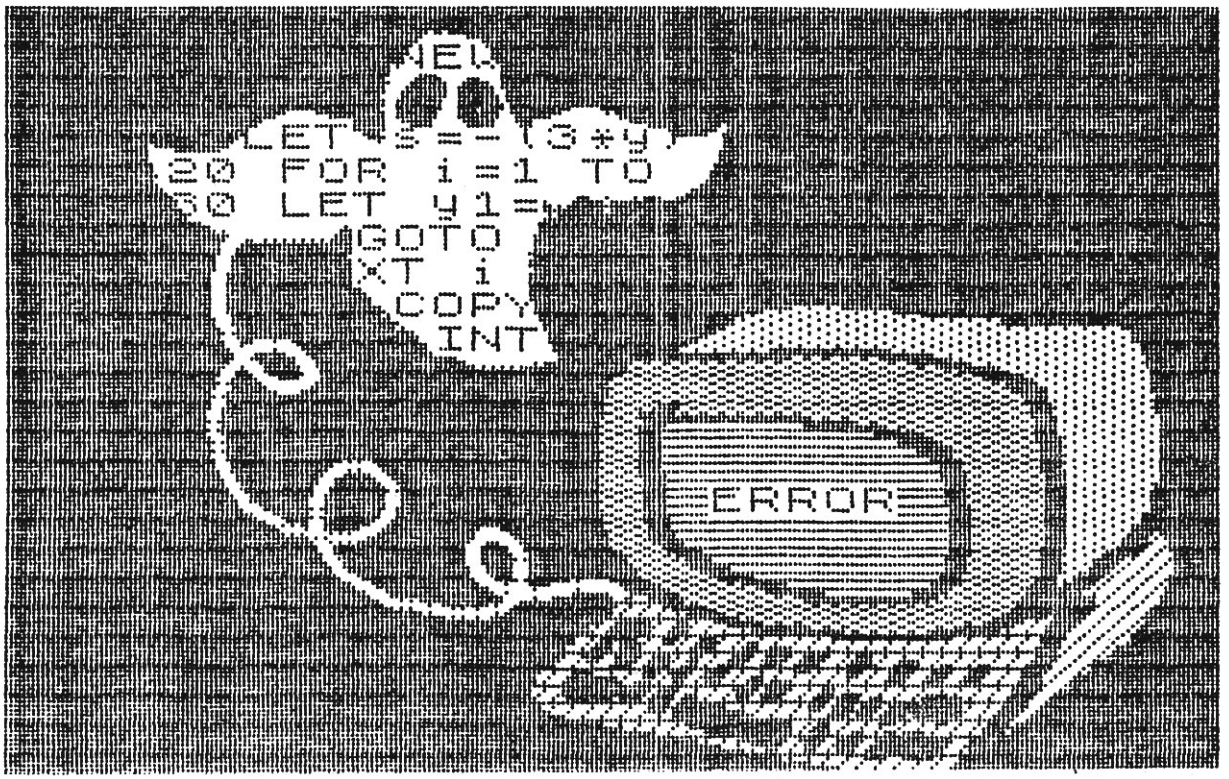
A *hardware* (ejtsd és magyar szövegben írd: *hardver*!) angol szó. Szó szerint fordítva kemény árut jelent. Egy angol úr vagy hölgy még harminc–negyven évvel ezelőtt is csak vasárut, vaskereskedést értett rajta. A „félrefordítás" történelme ugyan nem őrizte meg, de valószínű, a 20–25 évvel ezelőtti első magyarra fordított számítástechnikai szakcikkekben többször is előfordult teljesen érthetetlen módon a vasáru bolt. Mostanra azonban megtanultuk, mi mást is jelent ez a kifejezés. Ma így nevezik a különféle elektronikus áramkörökből és mechanikus berendezésekből felépülő számítástechnikai eszközöknek az összességét, magát a számítógépet, a köréje telepített, vele összekapcsolt adatbevivő billentyűzetet, adatmegjelenítő képernyőket és nyomtatókat, adattárolókat, sőt még az adatátviteli eszközöket is, egyszóval mindazt, ami kézzel fogható.

hardver

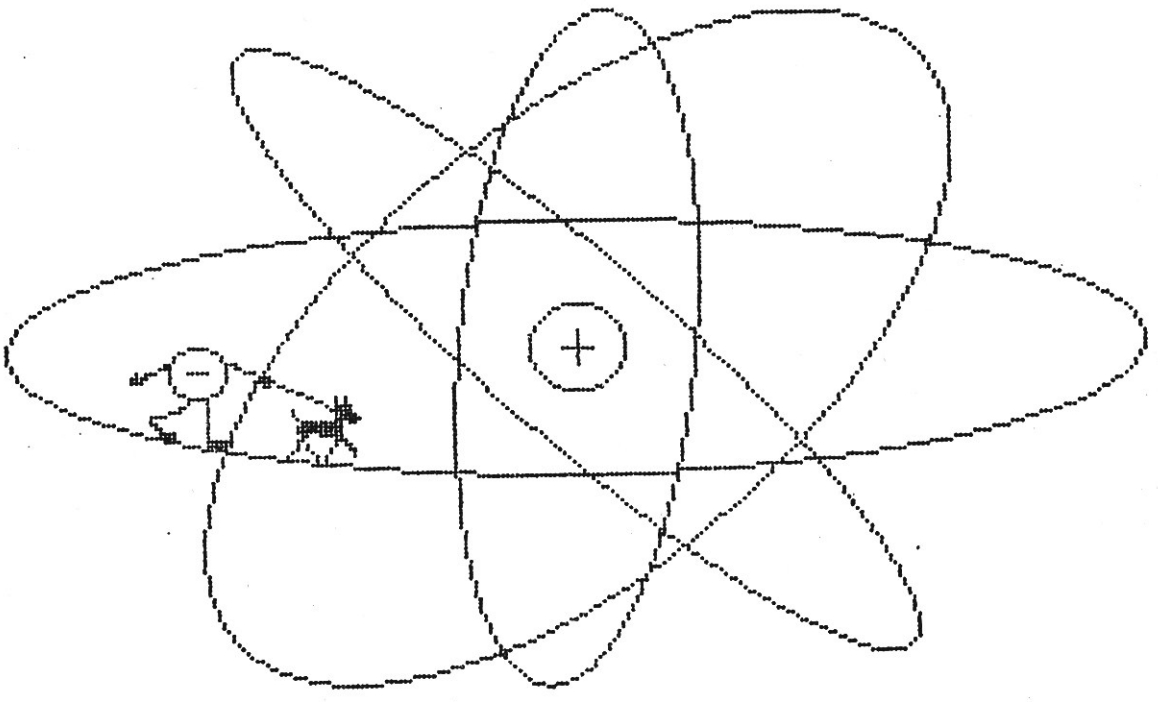
A *software* (ejtsd és írd: *szoftver*!) szó szerint lágy árut jelent. Ezzel aztán sem a hajdani urak, sem a hölgyek nem tudnának mit kezdeni. Ez a kifejezés tulajdonképpen csak a

szoftver

számítástechnika elterjedésével született. Ha a hardver a gép teste, akkor a szoftver a gép lelke. Ezen a programoknak az összességét értjük. Ez kelti életre a holt eszközt, ez teszi lehetővé, hogy a gép megoldja a rábízott feladatokat.



A gép lelke



Egy bit

Több kísérlet történt arra, hogy a hardver és a szoftver szavak helyett megfelelő magyar kifejezéseket honosítsanak meg. Sajnos egyik sem bizonyult életképesnek. A javasolt szavak vagy nem fedték le teljesen angol megfelelőjüket, vagy nem fogadta be őket a számítástechnikai köz- és szaknyelv. Így – jobb híján – most már a hivatalos nyelvőrök is engedélyezték fonetikus módon való írásukat és használatukat.

Azt, hogy egy számítógép műszaki szempontból mennyire jó, a hardver és a szoftver együttesen határozza meg. Az pedig, hogy egy gép milyen gazdasági hasznot hoz, nem csupán csak a gépnek az árától és a hardvernek és a szoftvernek az együttes teljesítőképességétől függ, hanem attól is, hogy hol és milyen feladatokat oldunk meg vele.

Könyvünk további fejezeteiben részletesen ismertetni fogjuk a legfontosabb hardver- és szoftverelemeket. Most ebben a bevezető jellegű fejezetben röviden összefoglaljuk a legfontosabb hardverjellemzőket.

Felépítésük, működésük, funkciójuk szerint a hardverelemek sokféleképpen osztályozhatók. Az egyik lehetséges csoportosítás a következő: processzor, központi vagy operatív tár (ezeket közösen — elsősorban a nagygépeknél — központi egységnek hívják), háttértároló, periféria és adatátviteli hálózat.

processzor, regiszterek, vezérlőmű, aritmetikai-logikai egység

A *processzor* tartalmazza azokat az áramköröket, melyek a *regiszterekben* és az operatív tárban tárolt adatokon elvégzik a különböző számítási és logikai műveleteket. Fő részei, melyekről a 9. fejezetben még bővebben szólunk majd, a *vezérlőmű* és az *aritmetikai–logikai egység.* Ez utóbbi nagyságtól függően általában 50—150 különféle műveletet tud elvégezni. Egy művelet végrehajtási ideje rendszerint egy tízezred (10^{-4}) és egy tízmilliomod (10^{-7}) másodperc között van. A 10—15 évvel ezelőtti gépek processzora egy egész szekrényt lefoglalt. A mai kisszámítógépek hasonló teljesítményű mikroprocesszorai egy csipetnyi chip-en, egy kb. 1/4 cm^2 felületű szilíciumlapocskán elférnek, és az áruk két-három nagyságrenddel kisebb lett, mint a hasonló teljesítményű régebbi processzorok ára.

operatív tár

A *központi* vagy *operatív tár* több tízezertől néhány millió, egyenként címezhető tárolóhely, melyek mindegyike egy jelet vagy adatot tartalmazhat. Bármelyik itt levő adattal a processzor közvetlenül tud dolgozni. A méretek és az árak kb. ugyanakkora mértékben csökkentek, mint a processzoroknál. Nagy teljesítményű számítógépeknél a műveletek felgyorsítása érdekében az adatoknak az operatív tárba történő be- és kiviteléről önálló vezérlőegység vagy egységek gondoskodnak.

tömegtárolók

A *tömegtárolók* óriási mennyiségű, több millió, sőt több milliárd jel tárolására alkalmas berendezések. Mivel az operatív tár kapacitása technikai és gazdaságossági okok miatt is korlátozott, a központi egységhez csatlakoztatott tömegtároló(ko)n tárolják azokat az adatokat, melyekre a feldolgozás folyamán szükség lehet, de az adott pillanatban éppen nem dolgozik velük a processzor, és az operatív tárba már nem férnek be. Amikor ezek az adatok kellenek, akkor ahhoz, hogy a processzor dolgozni tudjon velük, előbb be kell olvasni őket az operatív tárba. Az ehhez szükséges idő — beleértve az adatoknak a tömegtárolón való megtalálási idejét is — a leggyakrabban használt mágneslemezeknél általában néhány tized és század másodperc között mozog.

perifériák

A *perifériák* teremtenek és tartanak kapcsolatot a gép és a külvilág között. (Periféria: valaminek a külső része.) Ezeken keresztül vihetők be az ember által közvetlenül megadott adatok, és jelennek meg a számára közvetlenül hasznosítható információk is. Ide tartoznak többek között a képernyős adatmegjelenítők, billentyűzet, nyomtatók, különféle speciális jelzések értelmezésére, berendezések vezérlésére szolgáló egységek és régebbi típusú gépeknél a lyukkártya- és a lyukszalagbemenetek is.

adatátviteli hálózat

A processzor és a tár egymás közelében, esetenként egy chipen helyezkedik el. A perifériák vagy egymással kapcsolatban álló processzorok azonban különböző helyeken, egymástól akár több ezer kilométerre is lehetnek. Az adatoknak ezen helyek közti áramlását az *adatátviteli hálózat* biztosítja, mely az átvitelt biztosító közegen (telefonvezeték, üvegszál, rádióhullám) kívül magába foglalja az adatoknak a számítógépből a közegre helyező illetve arról levevő berendezéseket is.

A hardver egyes összetevőiről könyvünk második részében fogunk bővebben írni, míg a szoftver általános tulajdonságait a harmadik részben ismertetjük. A negyedik részben a szoftverhasználat különböző fogásait és félrefogásait írjuk le, míg az ötödik részben azt mutatjuk be néhány példán, hogy mire lehet a szoftvert és hardvert együttesen használni, vagyis mi mindenre jó a számítógép.

3.

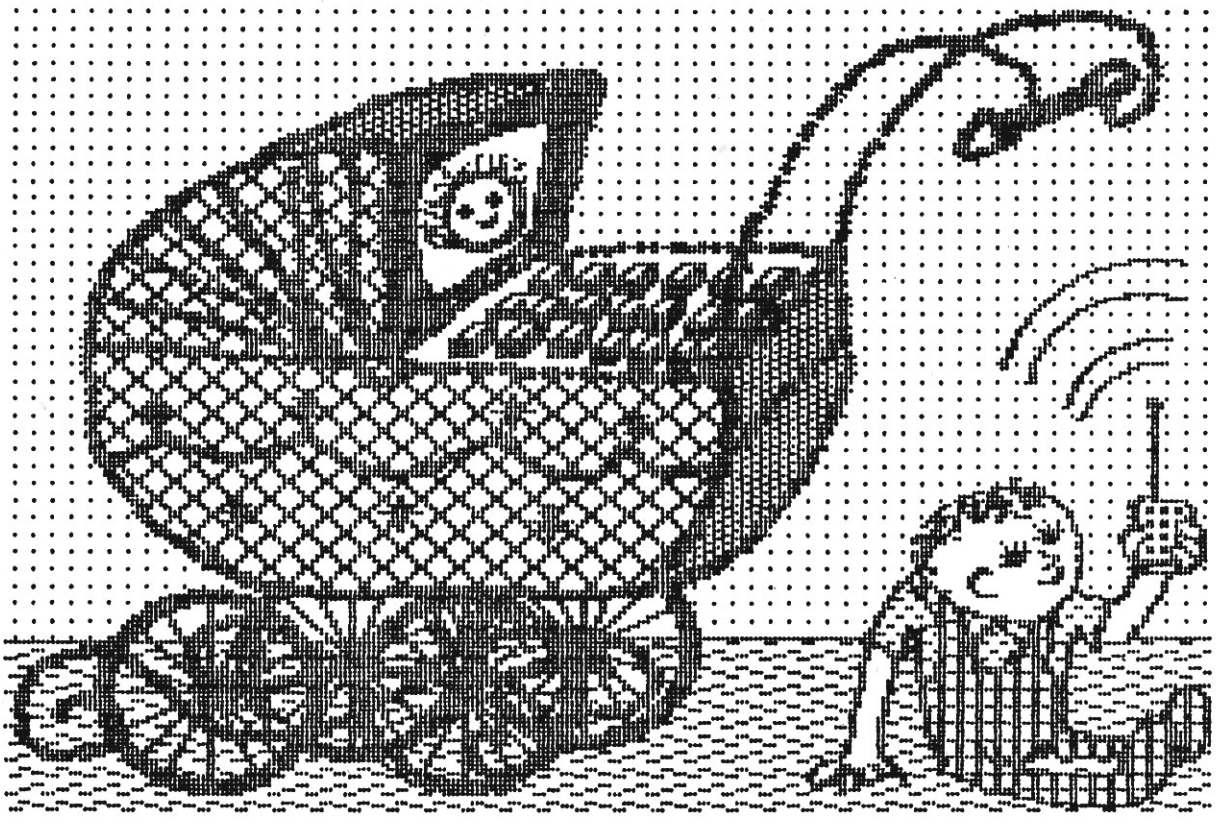
A MIÉNK MÁR EGY MÁSIK GENERÁCIÓ

(Számítógép-generációk)

Ki az, aki nem csodált még meg egy gyönyörűen karbantartott, a huszas vagy harmincas évekből származó veteránautót, amely fenséges eleganciával gördül végig a számára kijelölt útvonalon. Egyik-másik még rendszámmal és forgalmi engedéllyel is rendelkezik, de gazdája csak ritkán használja, mert a kocsi nagyon értékes, és igen sok benzint is fogyaszt. Az alig tíz éves számítógépeket viszont egymás után bontják szét, és selejtezik – jobb esetben néhány részüket múzeumba helyezik. Visszavonhatatlanul eljárt felettük az idő.

generációk

Pedig a számítógépeknek még nincs félévszázados múltja sem. Fejlődésük egyes jelentős szakaszait *generációk*nak nevezzük. Ez azt jelenti, hogy elvileg minden számítógép valamely generációba sorolható. A valóság azonban ennél bonyolultabb. Az egymást követő generációk ugyanis nem váltak el élesen: a fejlődés során készültek olyan gépek, amelyek

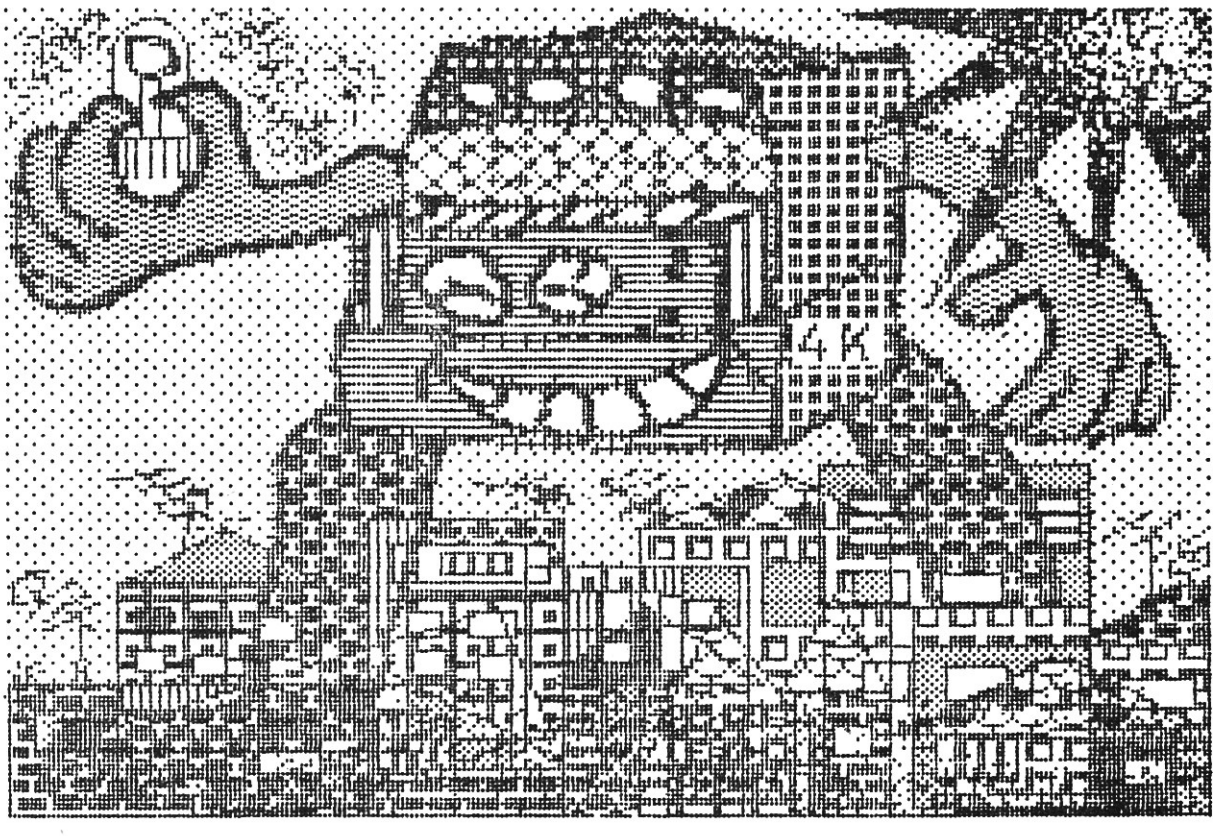


már a következő generáció egyes elemeit is magukban hordozták, de sok más tekintetben még az előzőhöz tartoztak. Egy-egy generáció objektív jellemzőit mindig utólag határozták meg, s az adott pillanatban senki sem fogalmazott úgy, hogy „akkor most készítsünk egy olyan számítógépet, amelyik minden szempontból a következő generációba sorolható." Ez még akkor is így van, ha több helyen olvashatjuk, hogy a japánok már az ötödik generációs gépek fejlesztésén dolgoznak, és a számítógépeket forgalmazó cégek legújabb típusaikat gyakran egy új generáció első tagjaként jelentik be, így próbálva minél nagyobb darabszámot eladni.

Éppen ezért a szakirodalom csak az első három generáció megítélésében egységes, és vannak olyan szakemberek is, akik egyenesen tagadják a generációkba történő sorolás létjogosultságát. Mindenesetre a különböző évtizedek számítógépei óriási mértékben különböznek egymástól.

„Tétova óriásokból" állt az *első generáció.* Ekkor még az áramköröket elektroncsövek alkották. Az üzemeltető mérnökök alapvető feladata a berendezés üzemképességének – mégpedig hosszú idejű üzemképességének (néhány óra) – biztosítása volt.

Ebben az időben maga a hardver jelentette a teljes számítógépet, azaz a gép „képességei" azonosak voltak a hardver „képességeivel", a csupa számjegyből álló úgynevezett gépi kód adta lehetőségekkel. A gépek nagyok voltak. Például az 1946-ban épített ENIAC nevű óriás 18.000 elektrocsőből, tízezer kondenzátorból, mintegy 70 ezer ellenállásból és 6 ezer kapcsolóelemből állt. Súlya megközelítette a 3 tonnát, terjedelme a 85 köbmétert. Egy óra alatt annyi energiára volt szüksége, amennyit 2000 darab 75 W-os égő fogyaszt el. Ugyanakkor ez a monstrum percenként mindössze 30 számítási műveletet tudott csak elvégezni. Természetesen ekkora alkatrészmennyiség esetében nagyon nagy volt a valószínűsége annak hogy egy vagy több elromlik. Ezért voltak „tétovák". Óriás mivoltuk pedig csak a terjedelmükből adódott. A negyvenes évek közepén készültek az első egyedi darabok, sorozatgyártásuk az ötvenes évek elején kezdődött el. Ilyenek voltak a hazánkban a hatvanas évek elején megjelent első gépek (URAL2, M3).

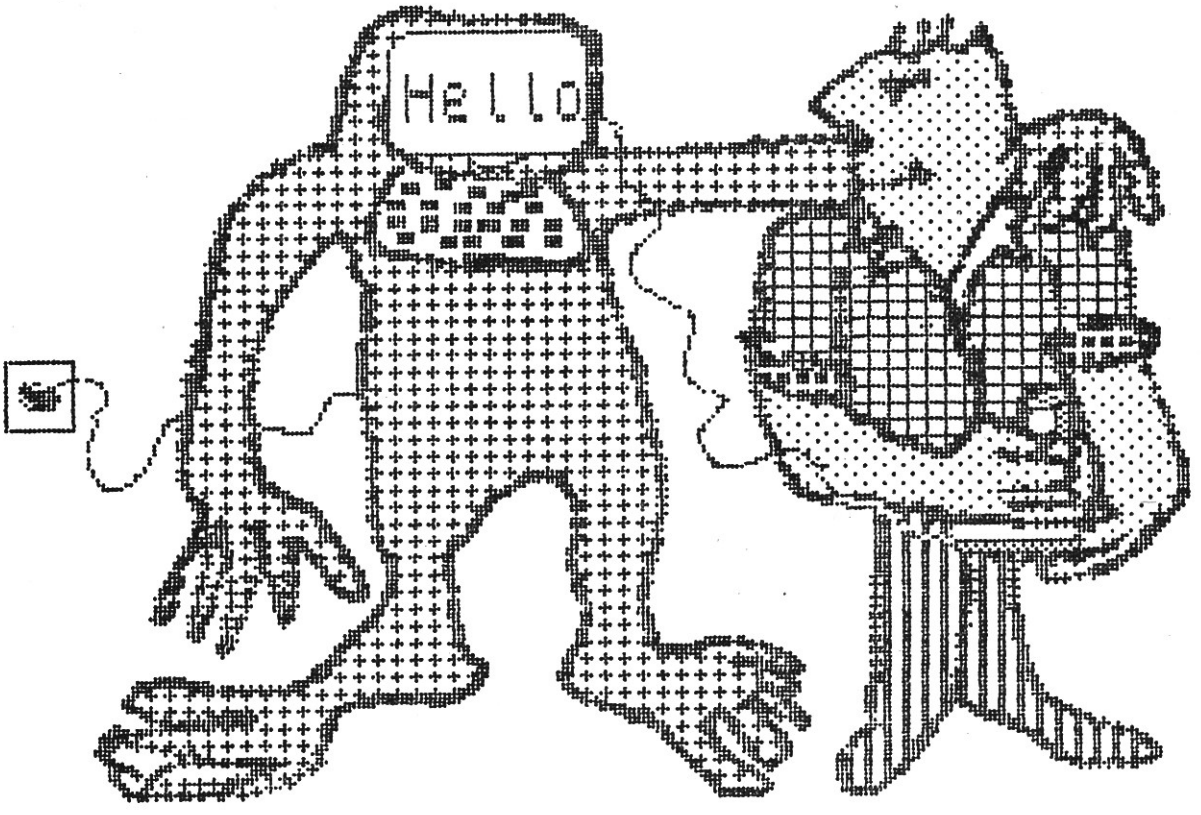


Tétova óriás

A tranzisztorok megjelenése nemcsak a híradástechnikában, hanem a számítógépépítésben is jelentős változásokat hozott. Az első tranzisztoros rádió 1954-ben, az első ilyen számítógép 1958 táján készült el. A tranzisztorbázisú „értelmes nagyok" alkotják a második generációs gépeket. Üzembiztonságuk már lényegesen javult, miközben a gépek méretei is számottevően csökkentek. Megjelentek a magas szintű programozást biztosító fordítóprogramok és a gép működtetését segítő úgynevezett monitorprogramok. Mai fogalmaink szerint még mindig nagyok voltak, de már értettek néhány nyelvet (Fortran, Cobol, Algol). Hazánkban is több második generációs gépet telepítettek (pl. ICT 1900).

A következő, immár harmadik generációt alkotják az „okos központok". Létüket az integrált áramkörök megjelenése tette lehetővé, és a szoftver súlyának további növekedése jellemzi. Ezekhez a gépekhez már igen jó hatékonyságú operációs rendszerek (lásd: 31. fejezetet!) tartoznak. Minderre a hetvenes évek elején került sor. A harmadik generációs számítógépeknél jelentek meg a hálózatok, amelyeknek lényege, hogy több (adott esetben több száz) felhasználó egyidejűleg használhatja a számítógépet. A felhasználók termináloknál (adatállomásoknál) ülve egy billentyűzet és egy képernyő segítségével tartanak kapcsolatot a központi távfeldolgozó géppel. Egyes rendszerekben több számítógép van összekapcsolva, így földrajzilag egymástól távol lévő felhasználók is kapcsolatba kerülhetnek egymással. Ezeket számítógép-hálózatnak nevezzük. A különböző helyen lévő gépek kapcsolata nem ritkán műholdakon keresztül valósult meg. Életciklusuk a hetvenes évek elején kezdődött. Jellegzetes, hazánkban még ma is megtalálható képviselőik az IBM 370 gépcsalád tagjai.

A *negyedik generációs* számítógépek már a jelent képviselik. Ők a „barátságos emberszabásúak". Talán azért, mert velünk élnek, nehéz meghatározni azokat a jegyeket, amelyek ennek a generációnak a jellemzői. Mindenesetre kimondhatjuk, hogy a félvezetős technikában tapasztalható miniatürizálási folyamat eljutott arra a pontra, amikor már minőségi változások következnek be. Ma egyetlen chipen sikerül megvalósítani a processzor funkcióit (ez a mikroprocesszor, minden személyi számítógép legfontosabb eleme).



Barátságos emberszabású

Ily módon egy processzor már bármilyen más eszközbe (gépbe, berendezésbe) beépíthető, ami a használhatóság körét többszörösére növeli. Nyilvánvaló, hogy pár éven belül mindez vissza fog hatni a szoftver területére. Egyre több szoftver „hardveresedik", azaz nem végrehajtandó program, hanem miniatűr áramkör formájában létezik. Ez teszi lehetővé, hogy ezek a gépek nagyon barátságosan viselkednek a felhasználóval szemben, vagyis nagyon kényelmes használatot biztosítanak.

Eközben a technológiai kutatások már az *ötödik generáció* felé haladnak. Az ötödik generációt a „mesterséges intelligencia" jelzővel illeti a szakirodalom. *Mesterséges intelligencián* azoknak a tulajdonságoknak az összességét értjük, amelyek alkalmassá teszik a számítógépet (esetleg egy számítógéppel vezérelt másik gépet) arra, hogy egy feladatot úgy hajtson végre, mintha azt ember végezné, mégpedig esetleg öntanuló módon is. (Az öntanuló mód azt jelenti, hogy a számítógép korábbi eredményei vagy eredménytelenségei, vagyis „tapasztalatok" alapján önmagát képes korrigálni.)

ötödik generáció,

mesterséges intelligencia

A mesterséges intelligencia fogalmába tartozik pl. a vizuális felismerés, a hangos ki- és bemeneten alapuló ember–gép kommunikáció, a természetes (tehát nem e célra konstruált) nyelvű programok alkalmazhatósága. Egyes területeken már jól alkalmazható részeredmények is vannak (pl. éppen a vizuális felismerés területén). Az Egyesült Államok piacán a mesterséges intelligenciával kapcsolatos eszközök forgalma 1984-ben 148 millió dollár volt. 1990-re ezt az adatot a szakértők 2860 millió dollárra prognosztizálják. A hat év alatt tehát közel húszszorosára várt érték már önmagában is jelzi, hogy mekkora fejlődés előtt áll a számítástechnika. Ehhez legfeljebb annyit lehet hozzátenni, hogy a számítástechnika területén az ilyen jellegű prognózisokat a valóság eddig mindig felülmúlta.

4.

MANCIKA MINT SZÁMÍTÓGÉP

(Az adatfeldolgozás főbb folyamatai)

A korszerű adatfeldolgozás nem új találmány, már a számítógépek előtt is létezett. Legfeljebb ma mást tekintünk korszerűnek, mint 30, 100 vagy 1000 évvel ezelőtt. Az emberi gondolkodás és a gépi adatfeldolgozás legfontosabb lépései ugyanazok. Az ember érzékszerveivel észleli a külvilágnak számára akkor éppen fontos eseményeit, jelzéseit, és ezeket eljuttatja az agyba. A számítógépnél ez megfelel az adatok bevitelének. Az agy az ember korábban összegyűjtött tudása, tapasztalata alapján feldolgozza az észlelt információkat. Ezt csinálja a számítógép is a tárában tárolt program és az esetleges korábbi feldolgozások eredményeként megőrzött adatok segítségével. Ez utóbbit az emberi emlékezőtehetséggel állíthatjuk párhuzamba. Végül az eredményt az ember megjegyzi, elmondja, vagy leírja. Ez a gépi feldolgozásnál az eredmények kivitele.

input

De nézzük meg ezt egy kissé részletesebben is, mégpedig egy olyan mindennapos feladatnak a megoldását követve nyomon, amelyet a bérszámfejtés üdvöskéje, Mancika és a számítógép egyaránt el tud végezni.

Első lépés: az adatbevitel. Mancika a jelenléti ívekről és a teljesítménylapokról összeírja, ki hány órát és mit dolgozott. Számítógépnél az adatrögzítő bebillentyűzi (vagy lyukkártyára lyukasztja és beolvastatja) a gépbe ezeket az adatokat. (Idegen, de a szakmában nagyon elterjedt szóhasználattal ezt *input*nak nevezik.)

Második lépés: a feldolgozás. A dolgozókról egyénenként összegyűjtött adatokból Mancika a vállalat elszámolási szabályai szerint kiszámolja a munkabért, s az egyéb előírá-

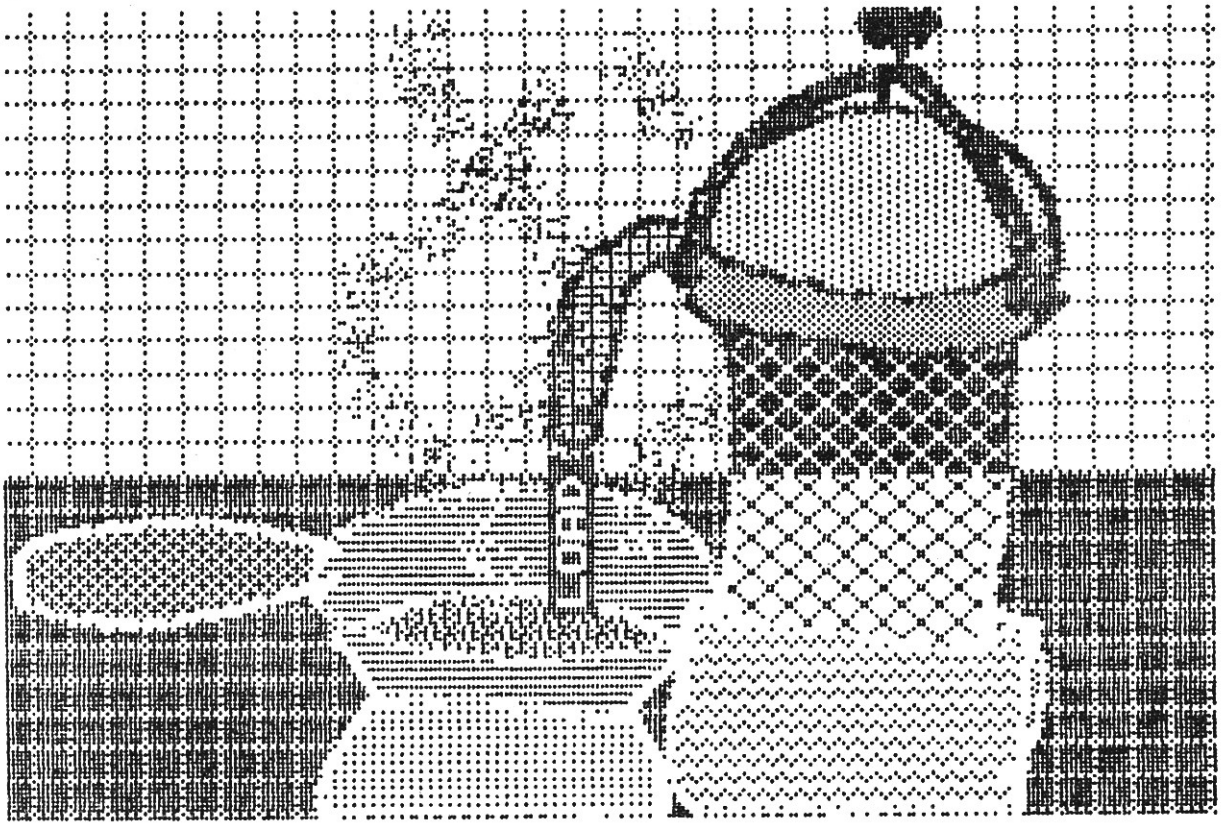
soknak megfelelően levonja belőle a nyugdíjjárulékot, az OTP-tartozást stb., és hozzáadja a családi pótlékot, meg az egyéb juttatást, ha ilyenek vannak. Mivel mindenkinek minden adatát nem tudja fejben tartani, munkájához a teljesítményelszámolási előírásokon kívül az egyes emberek személyi kartonját is előveszi. A számítógépen Mancika agya helyett a központi tárban levő bérelszámoló program dolgozik. Ez tudja a bérszámfejtésre vonatkozó összes előírást. Csak a gépnek ezt nem a főnöke magyarázta el, hanem a programozók töltötték az agyába, pontosabban a tárába.

Azok az adatok, amelyeket Mancika a kartonokból, könyvekből olvas ki, a számítógéphez kapcsolt háttértárolókon vannak és a program hívja be őket a tárba, amikor szükség van rájuk. A megfelelő számok összeszorzását, kivonását stb., amit a bérelszámoló fejben, papíron, asztali vagy zsebszámológéppel csinál meg, a gép aritmetikai egysége végzi el a program által megszabott sorrendben.

Harmadik lépés: az eredményszolgáltatás. Mancika a számolások elvégzése után végül legépeli a fizetési jegyzéket, s a pénztárnál ennek alapján fizetik ki a béreket. A számítógép is ugyanezt teszi, de az is lehet, hogy azonnal csekket állít ki, vagy közli annak a banknak a számítógépével, ahol az illető a pénzét tartja, hogy mekkora összeget írjon jóvá a számláján. (Ezt idegen szóval *output*nak nevezik.)

output

Látjuk, hogy a számítógép — legalábbis a bérszámfejtés folyamatában — lényegében ugyanazt csinálja, mint Mancika. Mi szól akkor mégis a gép mellett? Először is az, hogy



sokkal gyorsabb. Míg Mancika mondjuk átlagosan öt perc alatt végez egy személlyel, a gép ugyanennyi idő alatt akár 100 ügyet is elintéz. Másik előny az, hogy a gép nem téved. Ha helyes adatokat kapott, akkor mindig hibátlan eredményeket szolgáltat, míg Mancika, ha megzavarja a telefon csörgése vagy elábrándozik, fölcserélhet két tételt a normakönyvből, vagy rosszul írhatja ki a levonandó gyermektartás összegét.

Mi magyarázza akkor azt, hogy Mancika, aki fiatal, csinos és állandóan mosolyog, még mindig ott van a bérszámfejtésben? Talán az, hogy még a legkorszerűbb számítógép sem tud jó kávét főzni. De erre a kérdésre különben sem a számítástechnikai szakirodalomban kell magyarázatot találnunk.

5.

AZ ÉN GÉPEM, AZ ÉN IGEN KIS GÉPEM

(Számítógépek osztályozása ár és teljesítmény szerint)

A nyolcvanas évek közepére a számítógépek óriási kavalkádja alakult ki mind a gyártás, mind a felhasználás területén. A számszerű növekedés lenyűgöző méretű, nemcsak a számítástechnikailag fejlett országokban, hanem hazánkban is. Az Egyesült Államokban 1986-ban előállított hardver összeértéke 754 milliárd dollár volt. A fejlődés mértékét jól mutatja, hogy 1970-ben Magyarországon a legnagyobb számítógépek központi tára 128—256 kbájt, a hozzájuk kapcsolódó lemezek összkapacitása 50—100 mbájt volt, ami összemérhető, sőt esetenként kevesebb is, mint a ma már háztartásokban is megtalálható személyi számítógépek tárolóképessége. Ugyanakkor a hetvenes évek ilyen méretű gépeinek ára 1 millió dollár körül volt, míg jelenlegi hasonló nagyságú, de már jóval korszerűbb, megbízhatóbb, nagyobb teljesítményű utódjaiké ennek mintegy ötödére, tizedére csökkent az infláció ellenére is. A növekedés ütemét jellemzi, hogy a számítógépgyártó ipar, a hardvergyártás termelési értéke 1978 és 84 között az árak rohamos esése ellenére megkétszereződött, és várhatóan ismét kétszeresére nő értéke az 1984 és 88 közötti időszakban is. A szoftvergyártás területén még nagyobb volt az előállított termékek, szolgáltatások értékének a növekedése.

A számítógépek rendszerezéséhez elég nehéz egyértelmű, egységesen értelmezett és mindenki által elfogadott vezérfonalat találni. A 3. fejezetben ismertetett osztályozás a különböző időszakokban működő számítógépeket sorolja különféle generációkba. Nekünk azonban most a körülbelül azonos időben üzemelő számítógépeket kell osztályozni. Az egységes csoportosításhoz felhasználhatjuk, hogy léteznek olyan hardverjellemzők,mint például a tárolókapacitás, műveleti sebességek, illetve ezeknek meghatározott kombinációi, amelyek alapján az osztályozás lehetséges lenne. Ezeket a jellemzőket azonban meglehetősen sokféleképpen lehet értelmezni. Másik rendező elv lehetne a piaci ár. Az árak azonban országonként igen nagy eltéréseket mutatnak, és ráadásul meglehetősen gyorsan változnak. (Szerencsére állandóan esnek, a személyi számítógép kategóriában 1985—87 között még Magyaroszágon is átlagosan a felére csökkentek.)

Mindezek ellenére a gyakorlatban meghatároztak — és többé-kevésbé elfogadtak — három olyan alapvető kategóriát, amelyek valamelyikébe a legfontosabb hardverparaméterek és az árak nagyságrendje alapján a különböző számítógép-típusok majd mindegyike besorolható. Ezek:

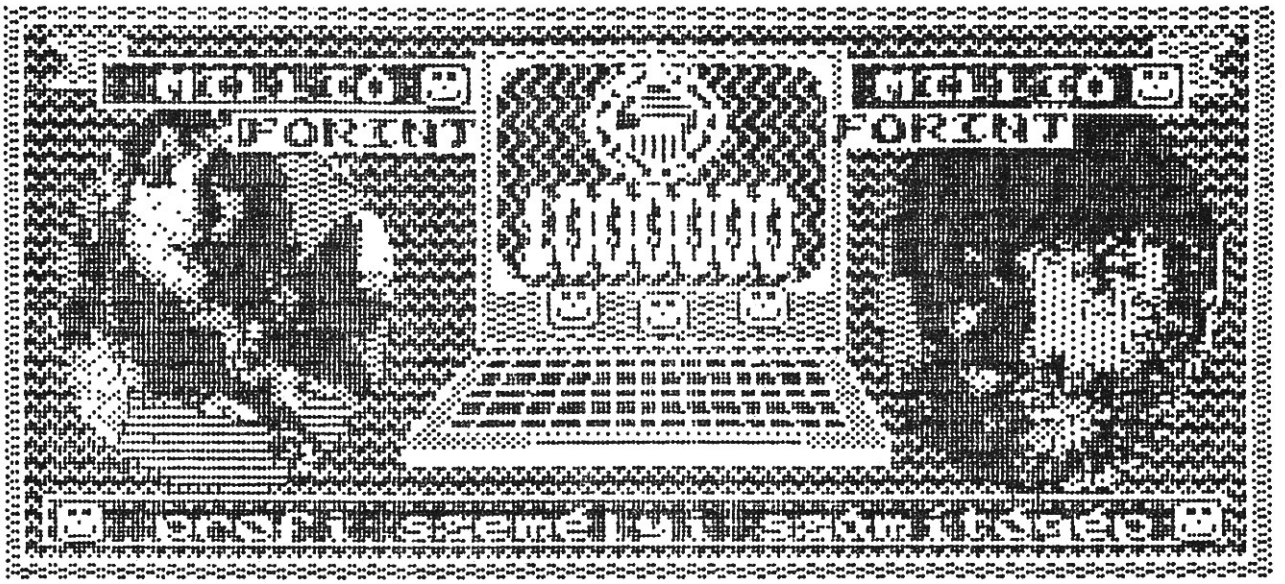
— a mikro-, vagy személyi
— a kis- vagy mini- és
— a nagyszámítógépek kategóriája.

(Korábban létezett még az úgynevezett közepes számítógépek csoportja is, de ma már ezek — elsősorban gazdasági okok miatt — „kihalóban" vannak. Szerepüket vagy a komoly kiépítettségű ún. mega-mini gépek, vagy a nagygépeknek a viszonylag kisebb kiépítettségű változatait vették át.)

E három csoport közül ebben a fejezetben részletesebben csak a mikrogépekkel foglalkozunk, mivel az olvasó leggyakrabban ezekkel találkozik, ezekkel kerül kézzel fogható

kapcsolatba. A teljesség kedvéért azonban a fejezet végén néhány szót szólunk majd a mini- és a nagyszámítógépekről is.

Személyi számítógépeknél (gyakran pc-nek emlegetik, az angol personal computer = személyi számítógép rövidítéseként) manapság az árhatárokat úgy szokták megvonni, hogy személyi számítógép az, ami a nyugati piacokon 100—5 ezer (esetleg 10.000) dollár, nálunk 10 ezer — 1,0 (2,5) millió Ft közötti összegbe kerül. Természetesen ezek a határok nem merevek. Egyrészt az azonos teljesítményű gépek árai rohamosan csökkennek, másrészt a személyi számítógépek teljesítménye egyre nagyobb lesz, így ugyanazon vagy némileg magasabb összegért sokkal több szolgáltatást nyújtanak. Ezáltal valósul meg a számítástechnikai paradoxon: minél olcsóbb lesz, annál többe kerül. Olyan helyeken ugyanis, ahol korábban elképzelhetetlen lett volna számítógép alkalmazása, ma az olcsóbb árak következtében gazdaságilag kifizetődővé vált. Sok kicsi sokra megy alapon végül összességében sokkal több pénzt költünk számítástechnikai berendezésekre és szolgáltatásokra, mint egy évtizeddel ezelőtt, amikor ezek ára még jelentősen magasabb volt. Az áresés oly mértékben begyűrűzött, hogy az elmúlt években többször leszállították a számítástechnikai berendezések vámtételeit. Ezáltal ezek ára nálunk a világpiaci árnak legfeljebb csak háromszorosa.

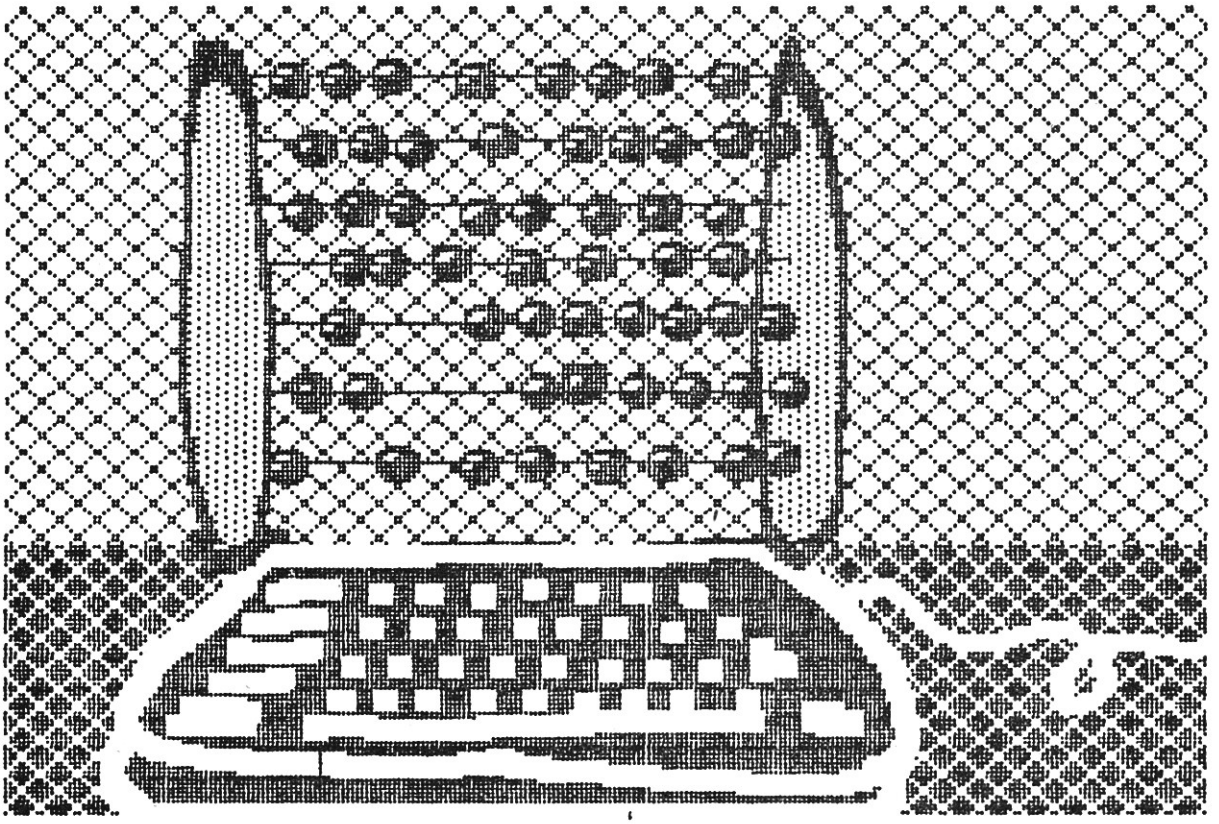


(A hazai számítógéppiac felépítése nem ilyen egyértelmű. Állami vállalatok korlátozott mennyiségben világpiaci áron is forgalmaztak személyi számítógépeket. Ugyanakkor az utasforgalomban behozott gépek ára helyenként torzabb arányt mutat.) Ennek ellenére a személyi számítógépek Magyarországon is egyre inkább tért hódítanak. Nemcsak kisebb vállalatok, intézmények, szövetkezetek vesznek ilyen gépet, hanem, ami a legfontosabb, szinte minden iskolában hozzáférhetnek a diákok az iskola saját kis személyi számítógépéhez.

Technikai szempontból a személyi számítógépek főbb jellemzői a következők:

- mikorprocesszoron alapulnak;
- operatív táruk általában 16—512 kbájt közt van;
- legalább egy tömegtároló mágneskazetta, hajlékony (vagy fix) lemez hozzá van építve, vagy kibővíthető vele;
- az ember—gép kapcsolat billentyűzeten és képernyőn keresztül párbeszédes formában megvalósítható. Igény esetén az eredmények kiírására nyomtató iscsatlakozik hozzájuk;
- legalább egy magasszintű nyelven (általában Basic) lehet rajtuk programokat írni;
- a felhasználók számára típusfeladatok megoldására kész programok állnak rendelkezésre.

Méretük (áruk és kiépítettségük) függvényében megkülönböztetünk *házi* és professzionális személyi számítógépeket. Az előbbiek alapkiépítettségben 100 $ (nálunk 10 ezer Ft) körüli áron kapható berendezések. Táruk aránylag kicsi, adatmegjelenítőként általában normál tévékészüléket, háttértárolóként gyakran csak kazettás magnetofont lehet hozzájuk kapcsolni. Különféle játékprogramokat és egyszerűbb otthoni feladatok megoldására szolgáló kész programokat árusítanak hozzájuk. A tulajdonos saját maga is készíthet igényeinek megfelelő új programokat, általában Basic nyelven. Tipikus képviselője ennek a kategóriának a Magyarországon gyártott PRIMO gép, külföldi gyártmányok közül a Sinclair Spectrum, valamint a már némileg magasabb árkategóriába tartozó Atari 800 XL, a Commodore C 64, C 16, plus/4 és az iskolákban elterjedt, szintén hazai gyártmányú HT 1080 Z, TV Computer (Videoton).



Iskola-számítógép

PPC

A *professzionális személyi számítógépek* ennél sokkal komolyabb hardverrel és szoftverrel rendelkeznek. A legnagyobbak, a számítástechnikai célú professzionális gépek lehetőségei, teljesítménye sok tekintetben eléri az egy évtizeddel ezelőtt még a közepes, sőt nagygépek kategóriájába tartozó gépek szolgáltatásait. Operatív táruk 512 kbájtra, vagy még efölé is bővíthető, Basicen kívül rendszerint programozhatók más magas szintű nyelveken, adatbázis-kezelő, szövegszerkesztő és különféle gyakorlati feladatok megoldására alkalmas szoftver kapható hozzájuk. Ezeknek működéséről, alkalmazásáról a III. és IV. részben fogunk majd bővebben beszélni. Lehetőség van e gépeknek egymáshoz vagy egy központi nagy géphez való csatlakoztatására, az ezekben tárolt adatok elérésére, és ezáltal tárolóképességük megsokszorozására.

A személyi számítógépekre jellemző még, hogy kiskereskedelmi forgalomban korlátozás nélkül könnyen beszerezhetőek. Ebben a tekintetben már nálunk is komoly előrelépés történt. A Skála áruházakban, az OFOTÉRT-nál, a Bizományi Áruházban egyes géptípusok gyakran kaphatók,és legalább féltucat hazai cég, kisszövetkezet is gyárt ilyen gépeket. Igazán elégedettek azonban csak akkor lehetünk majd, ha az árak reális mértékű csökkenése mellett megfelelő folyamatos kínálat is rendelkezésre áll.

A személyi számítógépek teljes mértékben a felhasználó ellenőrzése alatt állnak. A gép ott van az üzemeltető íróasztalán, és általában, legalábbis egy adott időben csak egy személy számára dolgozik. Mivel ezeknek a gépeknek az ára már nem magas, nem okoz problémát az sem, hogy ha esetleg órák hosszat tétlenek, vagy, bár gazdájuktól folyamatosan kapják a munkát, az ember reakciója olyan lassú, hogy a gép kapacitásának és sebességének csak egy töredéke van kihasználva.

Nem valósítható meg ez az idilli együttműködés a drágább, igényesebb üzemeltetési körülményeket kívánó kis- és nagyszámítógépeknél. Ezek általában önálló, légkondicionált gépteremben, saját, külön erre a célra kiképzett üzemeltető személyzettel működnek. A felhasználó terminálon vagy mikroszámítógépen keresztül kapcsolódhat hozzájuk. Az 53. fejezetben egy számítóközpontban tett látogatás kapcsán bemutatjuk egy ilyen, komolyabb feladatok megoldására szolgáló számítóközpont üzemeltetési rendjét. A kisgépekhez általában 16–25 terminál vagy mikrogép kapcsolható még hatékonyan. Az ehhez hozzákötött személyi számítógép felhasználója már csak kisebb feladatait oldja meg a mikrogépen, a komolyabb számítási kapacitást igénylő munkákat átadja a jóval nagyobb teljesítményű kisgépnek.

kis-számítógépek

A *kisszámítógépek* főbb jellemzői: Tárolókapacitásuk 500 ezer – 1 millió karakter. Műveleti sebességük másodpercenként kb. 1 millió. A háttérkapacitás milliárdos nagyságrendű. Az ilyen berendezések átlagára 50 ezer dollár körül van, vagyis a professzionális személyi számítógépek átlagárának tízszerese. Természetesen itt is nagyon nagy szórással (25 ezer dollártól 200 ezer dollárig).

A kisgépek kategóriájába sorolt számítógépek programokkal és programfejlesztési eszközökkel való ellátottsága kielégítőnek mondható. Ebbe a kategóriába tartozik pl. a magyar R11, TPA 1140-es család, a szovjet SZM4 valamint az Egyesült Államok-beli PDP 11, a Hewlett Packard/3000 stb.

nagy-számítógépek

A harmadik kategóriába a *nagyszámítógépeket* szokás sorolni. Ezek igen drága (fél millió dollár körüli átlagárú), 4–8 millió művelet/másodperc sebességű, 5–10 millió karakter tárolására alkalmas tárral rendelkező berendezések; amelyek háttérkapacitása elérheti a 10 milliárd bájtot. Üzemeltetésüket külön e célra szervezett számítóközpontok végzik. Nagyszámú (több száz) terminál kiszolgálására alkalmasak. Hazánkban csak néhány ilyen teljesítményű számítógép üzemel (Honeywell 6000, IBM 370 és 4300 valamint az ESZR sorozatok nagyobb gépei).

„szupernagy"

Létezik a világon néhány számítógép, amelyre a *„szupernagy"* jelző illik. Ezeket majd mindig valamilyen speciális célra használják. A hazai környezetnél maradva, az itthoni gépek egy részét „közepesnek" szokás nevezni. Ezek a gépek valahol az általunk mininek illetve nagynak nevezett kategória között vannak, árban, kapacitásban, sebességben egyaránt. Nevüknek megfelelően tehát „közepes" teljesítményűek. Az ilyen gépek jelenlétének oka, egyebek mellett az, hogy a hazai számítástechnikai alkalmazások pillanatnyi általános helyzete zömmel ilyen nagyságrendű gépeket igényel illetve tud eltartani.

6.

ÁGYÚVAL VERÉBRE?

(Milyen gépet válasszunk magunknak?)

Sikerült tehát az előzőekben rendet teremteni a számítógépek világában. A mikróktól a szupernagyig sorakoznak lelki szemeink előtt. De melyiket válasszuk, ha nekünk is szükségünk van egy gépre?

Ha valaki személyi számítógéphez akar jutni Magyarországon, több lehetősége is adódik. Felkeresheti azokat az áruházakat, szaküzleteket amelyek már foglalkoznak számítástechnikai berendezések forgalmazásával. Egyre több magyar gyártó hirdeti termékeit az újságokban. Kedvező vámtételek mellett lehet behozni egyes személyi számítógépeket külföldről is. Milyen meggondolások alapján vesz ma valaki személyi számítógépet?

Ha magánszemélyről van szó, valószínűleg a pénztárcája lesz a választást befolyásoló legfontosabb szempont. Ha ez a magánszemély nem csak hobby vagy játék céljára veszi a gépet, (mert például tagja egy számítástechnikai gmk-nak, és munkaeszközt akar venni), bizonyára tanulmányozni fogja, hogy mit tudnak a kínált gépek, s ezek közül melyik felel meg az igényeinek.

S végül, ha az illető egy vállalat vagy intézmény számára vesz gépet, pontosan tisztában kell(ene) lennie azzal, hogy a gépet milyen célok érdekében, milyen feladatok megoldására kívánják megvásárolni. Csak olyan gépet szabad megvenni, amelynek hardver- és szoftver lehetőségei képesek ezeket a feladatokat megoldani. (Ugyan kinek jutna eszébe hogy több 50 méterre hordó puskát vegyen, ha az a feladat, hogy 500 méterre kell ellőni egy ágyugolyót. Ma még sok helyen több személyi számítógép együttes alkalmazásával akarnak megoldani olyan feladatokat, amelyeket valójában csak egy nagy számítógép tud ellátni.)

Csakhogy — a hasonlatnál maradva — a puska esetében a dolog egyértelmű, mert a puskáról már a boltban kiderül, hogy mit tud. A számítógép bonyolultabb jószág, ezért előzetes tájékozódás nélkül nehezebb megállapítani a korlátait. A puska esetében viszonylag egyszerű a használat célja is. A számítógépnél ennek előzetes tisztázása sokkal összetettebb feladat. Az elmúlt évek hazai tapasztalatai azt mutatják, hogy az ilyen vásárlásoknál is gyakran kizárólag a vállalati pénztárca döntött.

Ennek bizonyítására csak egyetlen adalék. A Heti Világgazdaság 1985 november 30-i számában olvashattuk a következőket: „. . . egyes megvett kisgépek (pl. C 64) tulajdonosai a gépek alkalmazásakor súlyos sebesség- és kapacitáskorlátokba ütköznek. Képtelenek újabb jelentős összegeket befektetni, hogy valamilyen megoldással gépüket alkalmassá tegyék a feladat megoldására." Ne vonjuk le ebből azt a következtetést, hogy a gép a hibás.

A konfliktusok forrása — bármilyen furcsán hangzik — az első itthon is megjelent mikrogépek (elsősorban a Commodore 64) vállalati szemszögből nézve alacsony ára. Néhány százezer forint, vagy akár egy-két millió egy nagyobb vállalatnak nem okoz problémát, különösen ha gazdaságilag mérhető eredmények kecsegtetnek. S mire kiderül, hogy az olcsó gépnek híg a leve, addigra már nyakig ülnek a pácban.

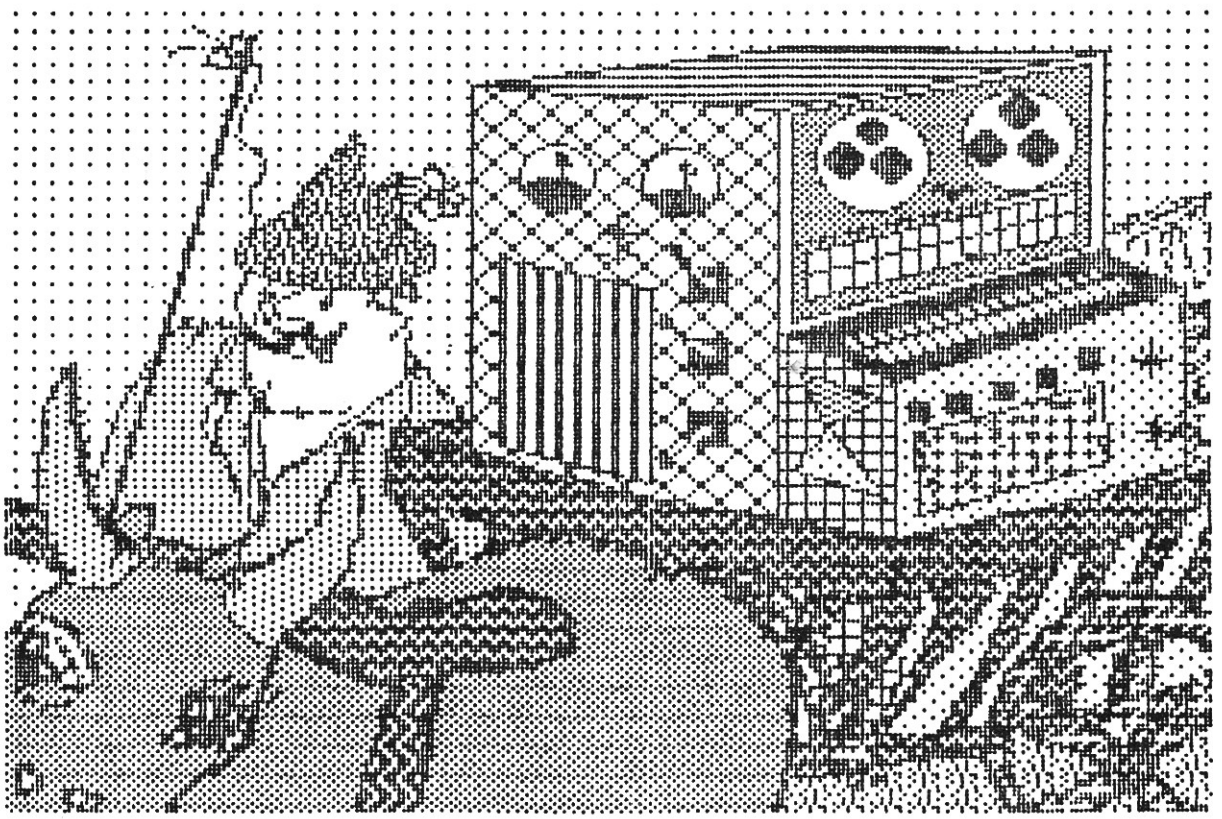
A dolgok megértéséhez gondoljuk végig, hogy miképpen fest a kép, ha egy vállalat hagyományos értelemben vett számítógéphez akar jutni.

Először is nincs ma hazánkban üzlet, amely ilyen gépeket árusítana, sőt olyan vállalat sem, amelyik komplett nagyszámítógépes rendszereket gyártana. A mini kategóriába tartozó gépek előállításával foglalkozik néhány hazai cég (pl. Videoton), de ezek sem termelnek „raktárra", hanem csak egyedi megrendelésre, előre rögzített termelési terv alapján. A nagyobb rendszereket külföldről kell beszerezni.

Természetes tehát, hogy ezt a lépést egy sor döntés előzi meg. Pontos felmérés készül, amelyben meghatározzák a vállalati számítógépesítési stratégiát, kijelölik az érintett területeket, körvonalazzák a géppel megoldandó feladatokat. E felmérés alapján határozható meg, hogy a beszerzendő gépnek milyen hardverparaméterekkel (tárméret, háttérkapacitás, műveletvégzési sebesség stb.) és szoftverrel (nyelvi fordítók, esetleg speciális alkalmazási programok, az igényeket kielégítő operációs rendszer) kell rendelkeznie.

További, úgynevezett járulékos költségeket is figyelembe kell venni:

- Létre kell hozni azt a környezetet, amelybe a számítógépet telepíteni fogják. Ezeknél a gépeknél – ma már inkább csak a mágneslemezegységek miatt – pormentes, állandó hőmérsékletű, megfelelően kialakított helyiségre (gépteremre) van szükség.
- A géphez jól képzett üzemeltető- és karbantartó személyzetet kell biztosítani.
- A gépterem mellett a személyzet elhelyezésére irodák, szociális helyiségek is kellenek.



Létre kell hozni a környezetet

Láthatjuk, hogy a hagyományos számítógép és egy mikroszámítógép beszerzése merőben más feladatot jelent a vállalat számára. Az előbbinél bonyolult az ügymenet: szakértők, építésztervezők stb. kellenek hozzá. A másikhoz jóformán elég egy főkönyvelői aláírás és egy anyagbeszerző. Ráadásul a személyi számítógép elhelyezésére elegendő egy irodai helyiség – amelyben mindössze a dohányzást célszerű megtiltani, és az átlagosnál valamivel nagyobb rendet illik tartani.

De azt is tudjuk, hogy ma már vannak mikrogépek, amelyek vetélkednek sok hagyományos számítógép képességével.

Ezért egy professzionális mikrogép beszerzése ugyanakkora hatással lehet egy kisebb vállalat információrendszerére, mint egy hagyonyos számítógép. Egyelőre a lehet szón van a hangsúly, mert a pillanatnyilag hazánkban elérhető professzionális személyi számítógépek egy-egy példánya nem képes a vállalati tevékenységek akkora területét érinteni, mint a hagyományos számítógép.

Mindez azonban nem jelenti azt, hogy több – esetleg hálózatba kötött – professzionális személyi számítógép ne lehetne mind műszaki mind gazdasági szempontból már ma is reális alternatívája egy nagygépnek – legalább is egy bizonyos feladat méretig, egy meghatározott műveleti sebességig és adatmennyiségig.

Kétségtelen, hogy egy-egy mikroszámítógép megjelenése a vállalati életben kisebb horderejű esemény, mint egy hagyományos számítógép beállítása. Ez az elmondottakból egyenesen következik. De a fejlesztési koncepció nélkül vásárolt mikrogépek porosodó irodabútorrá változnak, mert előbb-utóbb kiderül, hogy képességeik alapján nem alkalmasak az elképzelt feladatok megoldására. Még cipőkanállal sem férnek beléjük sem a feldolgozáshoz szükséges programok, sem az adatok.

A hagyományos számítógépeknél – éppen a magas beruházási és működtetési költségek miatt – jogos igény a „gazdaságosság", vagyis a ráfordítások megtérülése. A mikrogépeknél ilyen igénnyel nem szabad fellépni. (Még senkinek sem jutott eszébe azt mondani, hogy a Lánchíd nem gazdaságos, mert éjszaka nem megy át rajta elég autó.)

De ez nem jelenti azt, hogy a mikrogépeket ne képességeiknek megfelelő értelmes célra használjuk. Az értelmes használat előbb-utóbb eredményekhez vezet.

7.

KI MICSODA?
(Számítástechnikai munkakörök)

Az első generációs számítógépeket a felhasználók maguk programozták és üzemeltették. Szinte minden egyes program és gép egyedi termék volt. Így valójában nem is volt szükség kommunikációra és dokumentációra a programok készítésekor és végrehajtásakor. Ennek a módszernek modernizált változatát ma is gyakran megtalálhatjuk az interaktív rendszerfejlesztési munkáknál, illetve a személyi számítógépeknél. A feladat megfogalmazása, a munkák irányítása és ellenőrzése, a rendszertervezés és programozás mind a felhasználó kezében fut össze, és csak a gép tényleges működtetését végzi más, de még esetleg ezt is a felhasználó csinálja a saját személyi számítógépén, vagy irányítja a terminálról.

A második generációs számítógépeknél az üzemeltetést és a programozást olyan mértékben lehetett szabványosítani, hogy a harmadik generációs számítógépekhez már különböző, egyértelmű számítástechnikai munkakörök alakultak ki. Más csoport vagy más személy fogalmazza meg a feladatot, irányítja a fejlesztő–tervező munkát, készíti el a programokat, és végzi az üzemeltetés napi munkáját. Mint a régi gyermekmese mai változata: ez kitalálta, – ez megfogalmazta, – ez programozta, – ez üzemelteti, – és ez a pici mind megeszi (más szóval felhasználja az eredményeket).

A japánok által a századfordulóra tervezett *ötödik generációs* szuperszámítógépek felhasználásához már lehet, hogy nem szükséges több szakismeret, mint egy mai televíziós készülék működtetéséhez. Ezek a gépek szóban, a mindennapi beszédben használatos nyelven közölt utasításokat is megértik majd, (remélhetően nem csak a japánt), az emberi szemhez hasonló módon fognak látni, és az így érzékeltek alapján cselekszenek majd, sebességük és tárolókapacitásuk a jelenlegi csúcsteljesítményű gépeknél is több ezerszer magasabb lesz, magasszintű mesterséges inteligenciával, gondolkodó, tanuló, absztraháló képességgel fognak rendelkezni.

Addig azonban, amíg ez megvalósul, és elterjed a mindennapi gyakorlati munkában, még sok időnek kell eltelnie. A jelenlegi gépekkel csak úgy tudunk dolgozni, ha igényeinknek és a felhasználás szintjének megfelelő szaktudással rendelkezünk.

Igen sok felhasználónak csak meglehetősen homályos számítástechnikai ismeretei vannak. Meg vannak róla győződve, hogy a számítógép mindent tud, és mindent elvégez helyettünk. Mások kiváló számítástechnikai szakembernek és számítástechnikai ezermesternek hiszik magukat. Mindent saját maguk akarnak megoldani, függetlenül attól, hogy ez hatékony vagy gazdaságtalan gépkihasználást von maga után. E két szélső típus között van a felhasználók zöme, akik nem akarják átvenni a rendszerprogramozók munkáját, de azért elég számítástechnikai ismeretük és alakalmazási gyakorlatuk van ahhoz, hogy egyértelműen definiálhassák, sőt ha szükséges, elkészítsék a programokat valamilyen magas szintű programozási nyelven, vagy ki tudják választani a megfelelő programcsomagokat.

A számítógépes feldolgozási munkának – most már a gép oldaláról tekintve – különböző fázisai vannak. Mindegyik munkafázisnak megvannak a saját specialitásai, bár a különböző fázisok esetenként átlapohatják egymást, és széles ismeretekkel rendelkező

személyek szaktudása több munkafázisnak is megfelel. Az alábbiakban röviden ismertetjük a tipikus számítástechnikai munkaköröket és helyüket a rendszerszervezés, rendszerfejlesztés és alkalmazás folyamatában.

felhasználó

A *felhasználó* általában saját munkaterületének szakembere. Ez a terület lehet például közgazdaságtan, magfizika, nagy beruházások tervezése és irányítása, vagy bármilyen más szakterület, ahol a számítógép felhasználható. A fő feladata, hogy a munka kezdetekor pontosan és egyértelműen definiálja: mit kíván a számítógépes rendszertől, mikorra kell annak elkészülni, mennyi pénzbe kerülhet, és a munka végén az ő feladata az eredmények kiértékelése. Ha számítástechnikai ismeretei elegendőek, akkor a programozási munka jelentős részét el tudja készíteni egy magasabb szintű, vagy a feladatra orientált programozási nyelven, illetve le tudja írni a megoldáshoz szükséges lépéseket annyira részletesen, hogy ebből a programozó már el tudja készíteni a programot. A műszaki–tudományos felhasználók jelentős része e két kategória valamelyikébe tartozik. Bár számítástechnikai szemmel nem tekinthetők profi szakembereknek, és az általuk elkészített programok nem biztos, hogy a leghatékonyabban és a legbiztonságosabban működtetik a számítógépet, ez a legtöbb esetben mégsem jelenti az erőforrások összességének nem megfelelő hasznosítását. A kevésbé hatékony programozási munkát és gépi programot ellensúlyozza az egyszerűbb és gyorsabb rendszertervezés. Ez a tágabb értelemben már gazdaságos munkamódszer megmagyarázza, miért előnyös az interaktív programozás lehetővé tétele és a személyi számítógépek üzembeállítása. Az üzleti, vállalati, kereskedelmi és államigazgatási felhasználóknak rendszerint nincs elég idejük, hogy megtanulják a programkészítést, vagy ha meg is tanulták, akkor arra, hogy ezt ténylegesen maguk is csinálják. Ennek ellenére a számítóközpontokkal szemben támasztott igényeik legalább akkorák, mint a tudósok és mérnökök követelményei. Szerencséjükre a legtöbb esetben elegendő pénzük van ahhoz, hogy modellalkotókat, rendszerszervezőket alkalmazzanak a probléma specifikálására és programozókat a feladat programozására. Ennek következtében megjelentek az e feladatok elvégzésére kiképzett specialisták, akik aztán együttesen, csapatmunkával készítik el az egyedi célú illetve általánosan használható rendszereket. Összefoglalva a következő, speciálisan számítástechnikai fő munkaköröket különböztethetjük meg.

rendszerszervezők, folyamatszervezők, programtervezők, kódolók, rendszerprogramozók

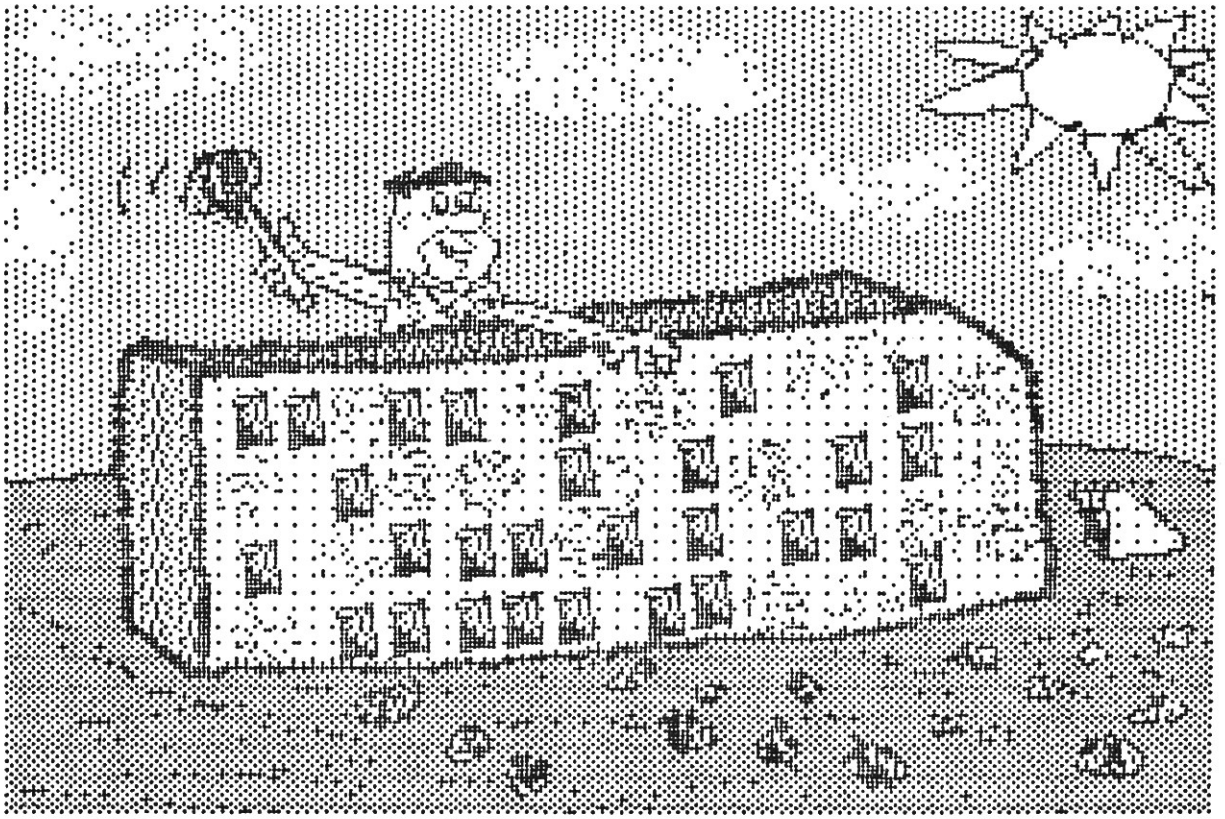
A *rendszerszervezők,* akik a felhasználók, vagy a rendszer logikáját, matematikai, műszaki gazdasági modelljét elkészítő modellalkotók, és a programozók közötti kapcsolatot hozzák létre. Elegendő szakismerettel kell rendelkezniük a felhasználó alkalmazási területén ahhoz, hogy ne csak megértsék a problémáit, hanem meg is tudják találni a leghatékonyabb módszert annak számítógépes megoldására. Feladatuk közé tartozik az adatok szervezése, és a bejövő bizonylatok és az eredményt szolgáltató táblázatok formájának meghatározása is. Nagy rendszerek egyes részfolyamatainak megszervezésében segítségükre vannak a *folyamatszervezők.*

A rendszerszervezők leírása alapján az *alkalmazási programozók* készítik el a számítógépes munkák végrehajtására szolgáló programok tervét és kódolását is. Ebből a szempontból különbséget tehetünk *programtervezők* és *kódolók* között. Az előbbiek a rendszerszervezők folyamatábrája alapján írják meg az egyes kisebb logikai egységekben végrehajtandó tevékenységek programjait. Gyakran előfordul, hogy az alkalmazási programozók fejlesztik ki azokat az új eljárásokat és programokat, amelyekkel a felhasználó a későbbiekben dolgozni fog.

A *rendszerprogramozók* feladata a rendszer általános üzemeltetéséhez szükséges szoftver létrehozása, karbantartása, ellenőrzése és bővítése. Fő céljuk a kialakítandó rendszer hatékonyságának növelése. Munkájukhoz tartozik még a tanácsadás az alkalmazói programozók részére rendszerhibák esetén. (Illetve a karbantartó mérnökök szidása, ha a hardver rossz, vagy a szoftver nem úgy működik, ahogy gondolták vagy elképzelték.)

Az *üzemeltető személyzet* irányítja és ellenőrzi az egész számítógép-rendszer működését. Ők veszik át a programozók munkáit, ütemezik a számítógépbe való bevitelét és az eredményeknek az illetékesekhez való eljuttatását. Gondoskodnak arról is, hogy a munkához szükséges adathordozók a gép megfelelő író/olvasó egységein legyenek, amikor szükség van rájuk. Külön részlegben dolgoznak ugyan, de az üzemeltető személyzethez számítjuk az *adatrögzítőket* is. Feladatuk az általában csak az ember által olvasható adatoknak a gép számára érthető, érzékelhető formában történő átírása.

adatrögzítők



A *karbantartó mérnökök* és a *műszakiak* fő feladata az, hogy biztosítsák a hardver működését.

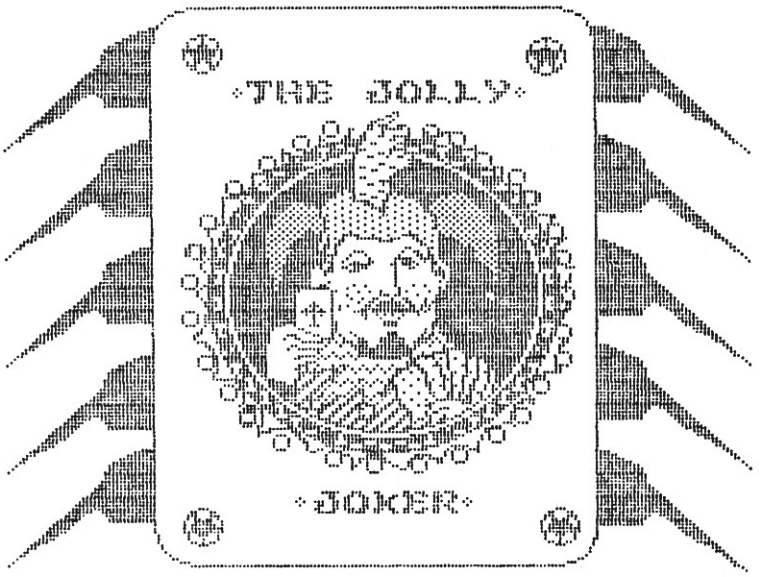
karbantartó mérnökök, műszakiak

A hatvanas évek közepére kialakult munkamegosztás a rendszerfejlesztő csoporton belül napjainkra bizonyos mértékig átalakult. Az emberi munka felértékelődése következtében egyre több szoftverfeladatot általános- vagy célhardverként beépítenek a gépbe. Ezenkívül a költségek további csökkentése érdekében az egyedi célú programok helyett igyekeznek minél több általánosan felhasználható szoftvert gyártani. Ily módon a rendszerprogramozók elébe mennek a felhasználók, modellalkotók igényeinek, és olyan speciális programokat hoznak létre, melyek segítségével a feladatok minimális programozási munkával, inkább a kész szoftverrendszernek a feladathoz való „méretre illesztésével" megoldhatók. Ilyen szoftvertermékek a különböző adatbázis-kezelő rendszerek (személyi, kis- és nagyszámítógépekre), táblázatkezelő, szövegszerkesztő és grafikus megjelenítő programok (elsősorban személyi számítógépekre), különböző irányítási feladatokat, műszaki–tudományos feladatokat megoldó programok (főleg kis- és nagygépekre). Néhányat ezek közül az ötödik részben ismertetni is fogunk.

A rendszerszervező és az alkalmazási programozó fő feladata ebben az esetben a meglévő, kész szoftver kiválasztása és a konkrét probléma megoldására történő adaptálása.

Tovább csökkent a munkakörök közti elkülönülés a személyi számítógépek alkalmazóinál. Mivel ezeknél viszonylag szerény programozási eszközök segítségével is sok feladatot eredményesen lehet megoldani, a felhasználó, a modellalkotó, a rendszerszervező, az alkalmazási programozó, a kódoló és az üzemeltető gyakran ugyanaz a személy, vagy a

csoport minden egyes tagja több funkciót is ellát. Ez természetesen nem jelenti azt, hogy a korábban felsorolt, önállónak tekinthető számítástechnikai munkakörök megszűnnek, legföljebb azt, hogy a köztük lévő elválasztóvonal nem marad olyan éles, mint volt néhány évvel ezelőtt.



A rendszer-szervező

Természetesen mindezek az emberek csak akkor tudnak dolgozni a számítógéppel, ha létezik – azaz rendelkezésükre áll a megfelelő hardver és az azt működtető alapszoftver. Az ezeket készítő tervező, fejlesztő és létrehozó, gyártó műszaki szakembernek, programozónak a tevékenysége már igen messze esik a rendszerszervezőétől, programozóétól.

A kapcsolódási pont csupán ott van köztük, hogy munkájuk eredményét, a számítógépet – annak minden műszaki és programozási adottságaival és korlátaival – használják fel a rendszerszervezők és a programozók a gyakorlati problémák megoldására.

A számítástechnikai munkakörökben foglalkoztatott szakemberek számát és a legfontosabb munkakörök közti megoszlását különböző statisztikai felmérések alapján megbecsülhetjük.

A következő táblázatban megmutatjuk, milyen növekedés várható 1982 és 1995 között azUSA különböző számítástechnikai munkaköreinél a foglalkoztatottak számát tekintve. A nagyarányú számszerű növekedés mellett szerkezetátalakulás is bekövetkezik. Az adatrögzítésre szakosodottak száma kissé, százalékos arányuk jelentősen lecsökken, mivel a személyi számítógépek és a kihelyezett terminálok széles körű elterjedésével az adatrögzítési feladatok jelentős részét a felhasználók, az ügyintézők közvetlenül elvégzik.

Év	1982		1995		1995/1982
Munkakör	e. fő	%	e. fő	%	
Szervező	251	21.9	467	26.3	1.86
Programozó	264	23.0	467	26.3	1.77
Gépkezelő	260	22.6	451	25.3	1.73
Adatrögzítő	319	27.8	286	16.1	0.90
Műszaki karbantartó	54	4.7	107	6.0	1.98
Összesen	1148	100.0	1778	100.0	1.55

A magyarországi számítástechnikai szakemberek számának alakulásáról, és ami ennél sokkal érdekesebb lehet, a jövedelmeikről a 37. fejezetben ismertetünk néhány adatot. Mire az olvasó odáig eljut, remélhetőleg már rendelkezik kellő ismerettel arról, hogy mit kell tudnia egy számítástechnikusnak. Ennek alapján azután azt is eldöntheti, érdemes-e ezt csinálni – ennyi pénzért. Ha csak az ott leírt számokat nem teszi már elavulttá a számítástechnika rohamos fejlődése vagy az időközben bekövetkező bérrendezések.

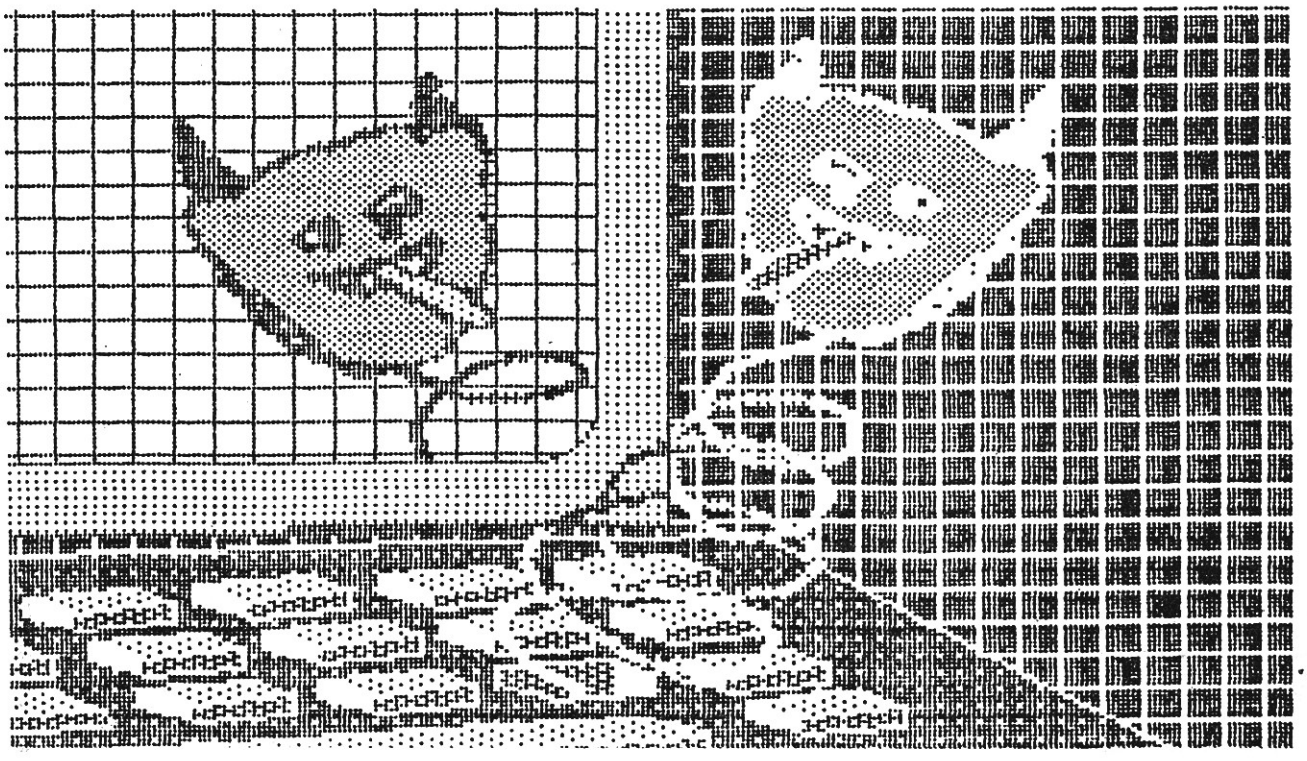
8.

SZÁMÍTÁSTECHNIKA GÖRBE TÜKÖRBŐL – AZAZ BELÜLRŐL NÉZVE

(Murphy (sz)ámítástechnikai törvényei)

Minden tudományágnak vannak örök igazságai. Itt most nem olyan korlátozott érvényű törvényekre gondolunk, mint például Newton mozgástörvényei, melyek a hétköznapi életünkben tökéletesen leírják a világot, de sem a relativisztikus sebességeknél, sem a mikrorészecskék világában nem állják meg a helyüket, hanem olyan ,,valódi alapigazságokra", mint amilyenekre például N. C. Parkinson alapozza a ,,politológia" tudományágát. A műszaki és természettudományoknak ezen görbe tükörből nézett, de a gyakorlati szakemberek által tapasztalati tények alapján teljes mértékben elfogadott tételeit Murphy fogalmazta meg legironikusabban, és ebből következően természetesen legpontosabban és legérthetőbben is. Maga az alaptétel igen rövid és tömör: ,,Ami el tud romlani, az el is romlik." Ebből az egyszerű axiómából aztán Murphy számos következtetést vezetett le. Ezek közül a legfontosabbak, melyek a számítástechnika területére is *egy az egyben* alkalmazhatók, a következők:

- Semmi sem olyan egyszerű, mint amilyennek látszik.
- Minden hosszabb ideig tart, mint ahogyan gondolod.
- Minden megoldás újabb problémákat szül.
- Ha egy berendezésben több dolog is meghibásodhat, közülük – és ebben bizonyos lehetsz! – az fog elromlani, amelyik a legnagyobb kárt okozza.
- Ha meggyőződtél arról, hogy egy bizonyos dolog csak négyféleképpen hibásodhat meg, és ezek mindegyike ellen biztos védekezési és elhárítási módot dolgoztál ki, akkor rendszered igen rövid időn belül felmondja a szolgálatot egy új, előre nem látott, ötödik fajta hiba következtében.



Murphynak igen sok követője akadt, és sok híve van ma is. Elméletét — melyet a tapasztalat a technika fejlődésével egyre nagyobb mértékben igazolt — sokan továbbfejlesztették. Ezek közül könyvünk megfelelő helyein bemutatjuk azokat, amelyek a hardver valamint a programozás területére vonatkoznak, és azokat is, amelyek a komplett számítástechnikai rendszerek működésképességére általánosított Heisenberg-féle határozatlansági-elvből következnek.

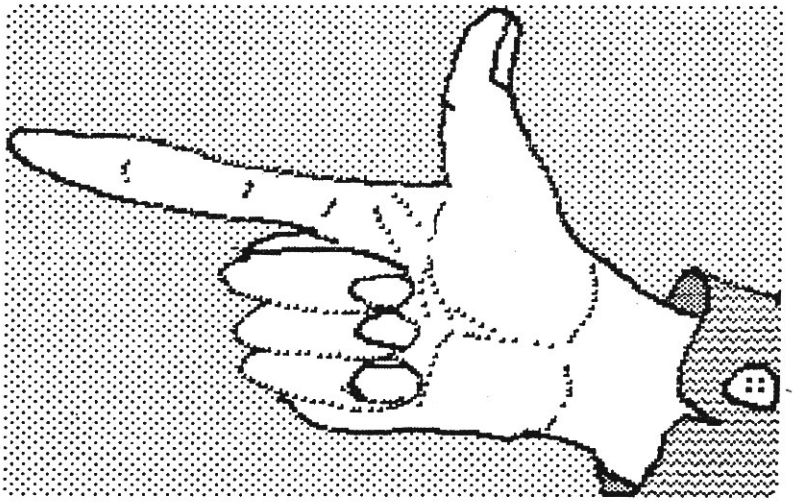
(Ennek az elvnek a korai, elektroncsövekkel működő számítógépekre alkalmazott következtetése volt az a tétel, mely szerint egy olyan berendezés, amelyben egyidejűleg négynél több elektroncsőnek kellene hibátlanul működnie, elvi okból sem üzemelhet. Matemaikai formában $dn \times dt \leqslant h$, ahol dn az 1 fölötti elektroncsövek száma, dt a hibamentes üzemmód időtartama másodpercekben, és h a számítások eredményét váró személy vércukorszintje két perccel a számítások befejezésére kijelölt határidő után.)

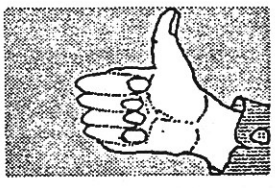
Ezen törvények és megállapítások egy részét a szerzők saját tapasztalataikból szűrték le, más részük — hasonlóan sok önálló szoftvertermékhez — idegen forrásokból származik. A továbbiakban azonban e törvények felismerőinek és megfogalmazóinak nevét nem tüntetjük fel. Maradjanak ismeretlenek, ugyanúgy, mint ahogy a legtöbb számítástechnikai hibának a valódi oka, sőt sokszor még a következménye is rejtve marad előttünk. Hogy azért mégis mit várhatunk Murphy elméletének számítástechnikus továbbfejlesztőitől, álljon itt erről az egyik ismert szaktekintély határozott véleménye:

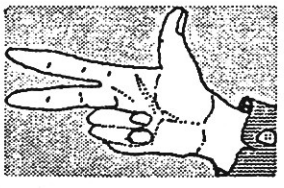
„Murphy egy javíthatatlan optimista."

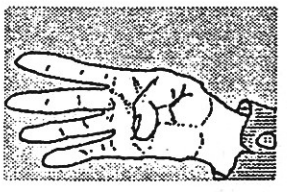
VÁGJUNK BELE !

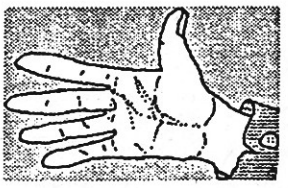
A HARDVER







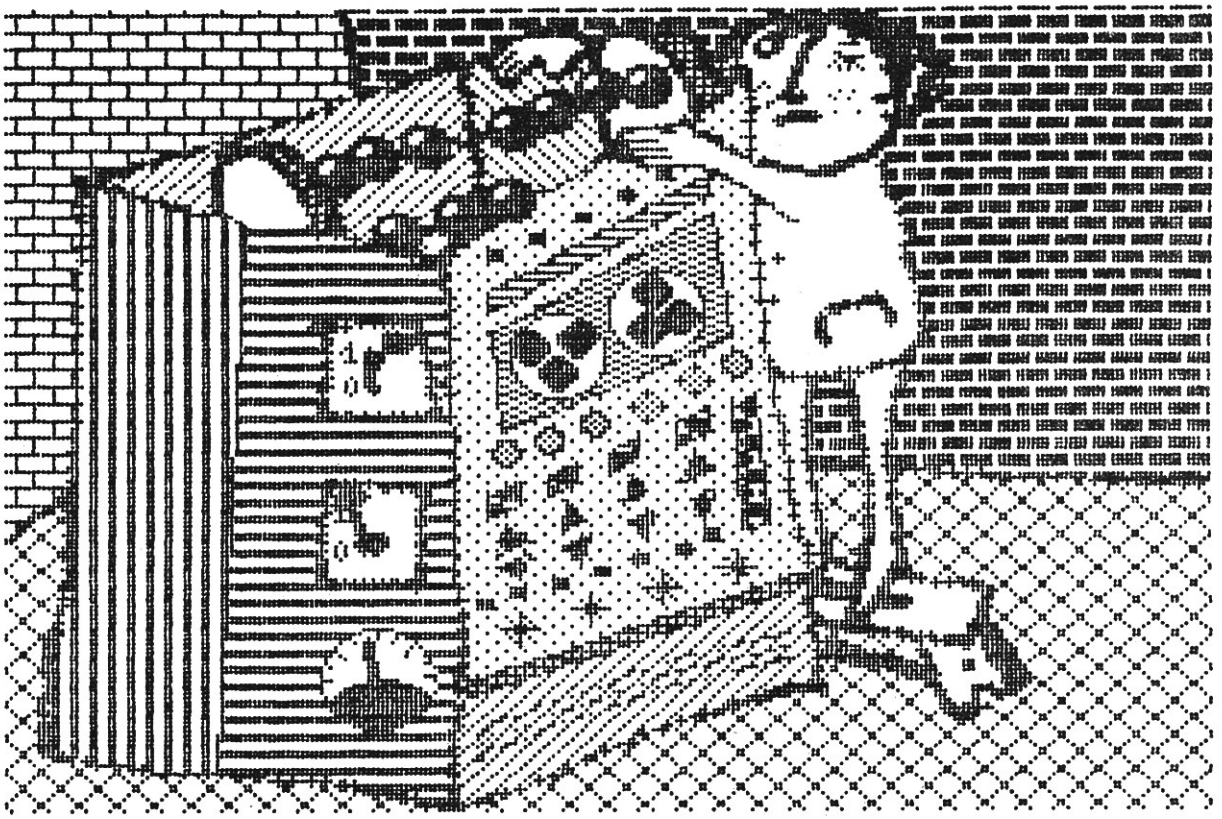




9.

A SZÁMÍTÓGÉPEK MŰKÖDÉSI ELVE

Ha a közlekedési eszközök sebessége az elmúlt 30 évben olyan arányban növekedett volna, mint a számítógépeké, méretük és teljesítményhez viszonyított áruk pedig hasonló mértékben csökkent volna, akkor ma a traktorok is gyorsabban száguldanának a szuperszónikus vadászgépeknél, a 2000 személyt befogadó luxus óceánjárók teljes térfogata beleférne egy valóságos méretű Trabant csomagtartójába, és a robogók egykori áráért még az infláció ellenére is több Mercedest lehetne kapni. (Persze, hogy ezeket az eszközöket mire lehetne használni, az merőben más kérdés.)



Ugyanakkor a technika ilyen mértékű fejlődése mellett a napjainkban általánosan elterjedt számítógépek működése még mindig ugyanazon az elven alapszik, mint amelynek alapján Neumann Jánosék több mint 40 évvel ezelőtt megépítették az első tárolt programú digitális elektronikus számítógépet.

központi processzor

A számítógép motorja, szíve és agya a *központi processzor.* Ennek fő részei az aritmetikai–logikai egység, a vezérlőmű, a központi tár vagy más néven operatív tár és a beviteli és kiviteli vezérlőegységek.

Az *aritmetikai–logikai egység* hajtja végre az összes számolási és logikai műveletet. Nagy integráltságú, igen gyors áramkörökből épül fel, melyekbe a műveletek operandusai, rész- és végeredményei kerülnek. Az áramköri elemek árának csökkenésével egyre több az olyan gép, amelyben a sebesség növelése érdekében egyes bonyolult utasításokat, utasítás-

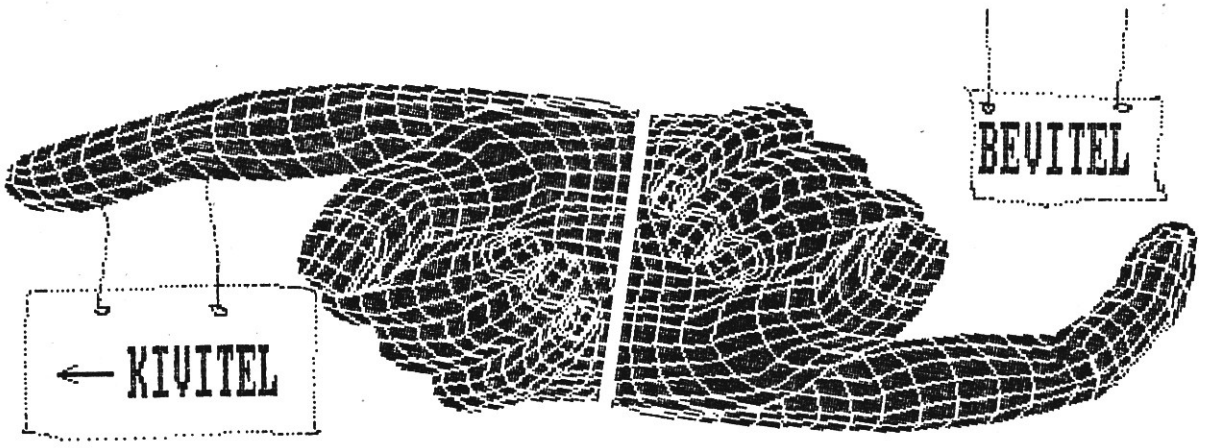
csoportokat csak az adott speciális célra szolgáló integrált áramkör hajt végre. Nagy gépeknél a különböző utasítások száma 150 körül van. A művelet végrehajtásának eredményességét egy külön regiszter tárolja. Ennek tartalma jelzi, ha a művelet nem volt végrehajtható, mert például az eredmény nagyobb (vagy kisebb) a gépben tárolható legnagyobb (vagy legkisebb) számnál, a két összehasonlítandó mennyiség közül az első vagy a második volt-e nagyobb,stb.

vezérlőmű

A *vezérlőmű* az ész. Ez ütemezi az egész számítógép működését. Ez szabja meg a program utasításainak végrehajtási sorrendjét. A soron következő utasítás címét, vagyis hogy hol található a központi tárolóban az utasítás, az utasításszámláló elnevezésű regiszterből veszi, melybe általában a központi tárolóban elfoglalt sorrendjük alapján kerülnek be egymás után az utasítások címei. Az utasításszámlálóban megadott címen talált bináris jelsorozatot a vezérlőmű utasításként értelmezi és ennek alapján parancsot ad az aritmetikai—logikai egységnek vagy a beviteli-kiviteli vezérlőegységnek az utasításban meghatározott művelet elvégzésére. Olyan esetekben, amikor valamilyen művelet eredményeként nem a közvetlenül soron következő utasítást kell végrehajtani, hanem máshonnan kell a programot folytatni, egy speciális utasítástípus, az ugróutasítás átírja az utasításszámlálóban levő címet, és a program ettől a helytől folytatódik.

operatív tár

A *központi-* vagy *operatív tár* tartalmazza a feldolgozandó adatokat és a program végrehajtásához szükséges utasításokat a gép által közvetlenül értelmezhető formában. Felépítéséről a könyv későbbi fejezeteiben majd részletesen beszélünk.



beviteli-
-kiviteli
vezérlőegységek

A *beviteli-kiviteli vezérlőegysége(ke)t* elsősorban nagyszámítógépekben alkalmazzák. Munkájukat, adatoknak az operatív tárból a bemeneti-kimeneti egységekre, a perifériákra való kivitelét, illetve onnan a tárba történő közvetlen behozatalát az aritmetikai—logikai egységektől függetlenül végzik. Így, a viszonylag hosszú ideig tartó adatbeviteli, és -kiviteli műveletek ideje alatt az aritmetikai—logikai egység ugyanannak, vagy más programnak a számolási és logikai műveletein dolgozhat. Az önálló beviteli-kiviteli vezérlőegységek működését a megfelelő programutasítás hatására a vezérlőmű indítja el, és ennek jelzik azt is, amikor az adatátvitelt befejezték.

10.

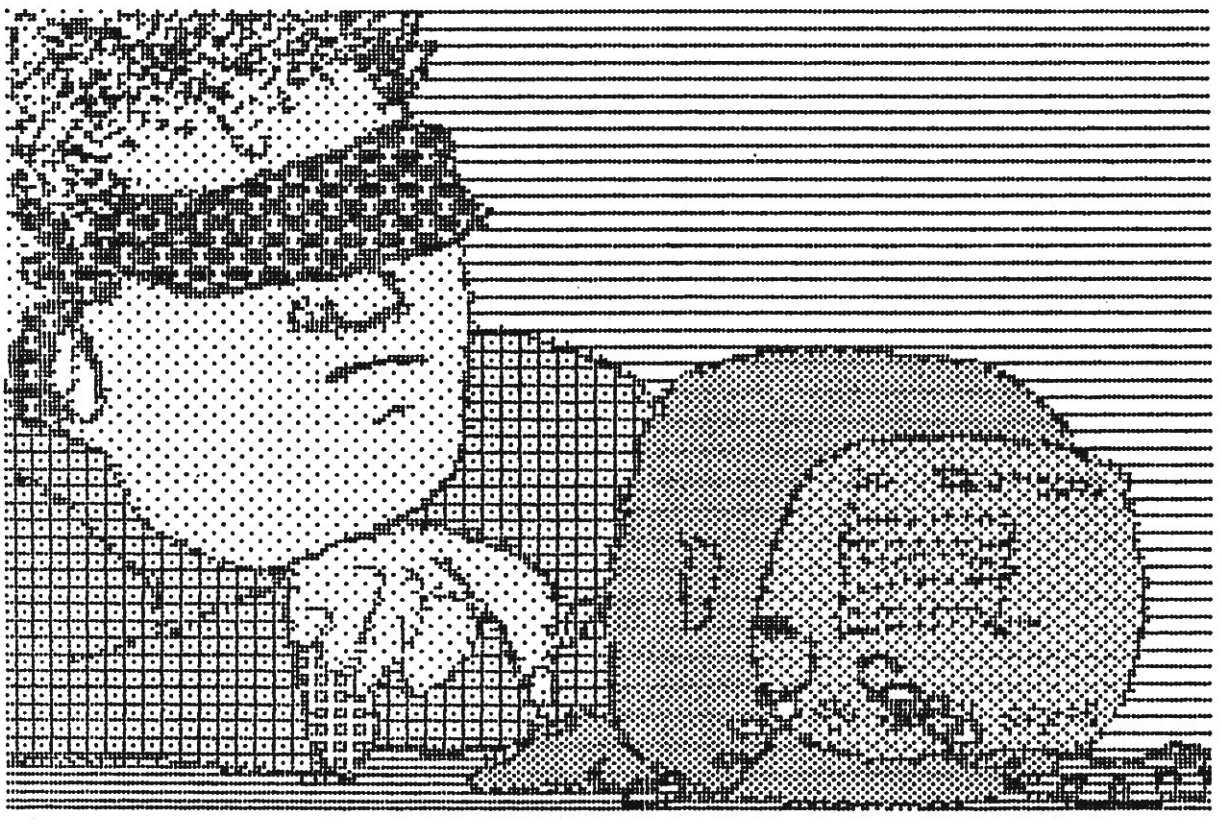
HOL TALÁLLAK?
(Bit, bájt, tárolócímzés)

A számítógép által tárolt és feldolgozott adatoknak az alapegysége a *bit*. A bitnek az értéke csak 0 vagy 1 lehet, amit a hardverben általában az áram folyása, vagy nem folyása, vagy a mágnesezettség egy adott vagy azzal ellentétes irányba való beállása jelez. A bit a gépben tárolható információ legkisebb egysége. A biteknek, az 1 és 0 számjegyeknek megfelelő hosszú sorozatával a számítógép számára minden információ leírható. A gépnek ahhoz, hogy a 0-tól 9-ig terjedő tíz számjegyet megkülönböztethesse egymástól, legalább négy bitre van szüksége. Ezekből a 0 és az 1 számjegyeknek tizenhat (2^4) különböző sorozata állítható elő. Ez hattal több mint amennyire szükségünk van, hiszen egy-egy sorozat egy-egy számjegyet képviselhet. Három bit azonban nem elegendő, ezzel csak nyolc (2^3) számjegyet tudnánk ábrázolni. Az angol ábécé nagybetűi hat bit felhasználásával ábrázolhatóak ($2^6 = 32$ különböző lehetőség), azaz minden betű egy nullákból és egyesekből álló jelsorozattal írható le. A számítógépekben rendszerint szöveges és numerikus adatokat is tárolnak, ezért célszerű ezeket egységes módon ábrázolni, természetesen úgy, hogy a gép különbséget tudjon tenni a betűk és a számok, valamint ezek egyes értékei között is. Ennek érdekében a számítógépben több bitet, általában nyolcat összevonnak egy nagyobb egységbe, ezt *bájtnak* nevezzük. Az egy bájtba foglalt nyolc bit – $2^8 = 256$ különböző jel sorozata – elegendő arra, hogy egy számjegyet, betűt, írásjelet, vagy akár egy gépi műveleti kódot leírjon. (Sőt, szöveges adatoknál így már különbséget tudunk tenni az ékezetes kis- és nagybetűk között és különböző írásjeleket is elhelyezhetünk.)

bit

bájt

A számítógépek legtöbb művelete egy bájtra vagy bájtok sorozatára vonatkozik.



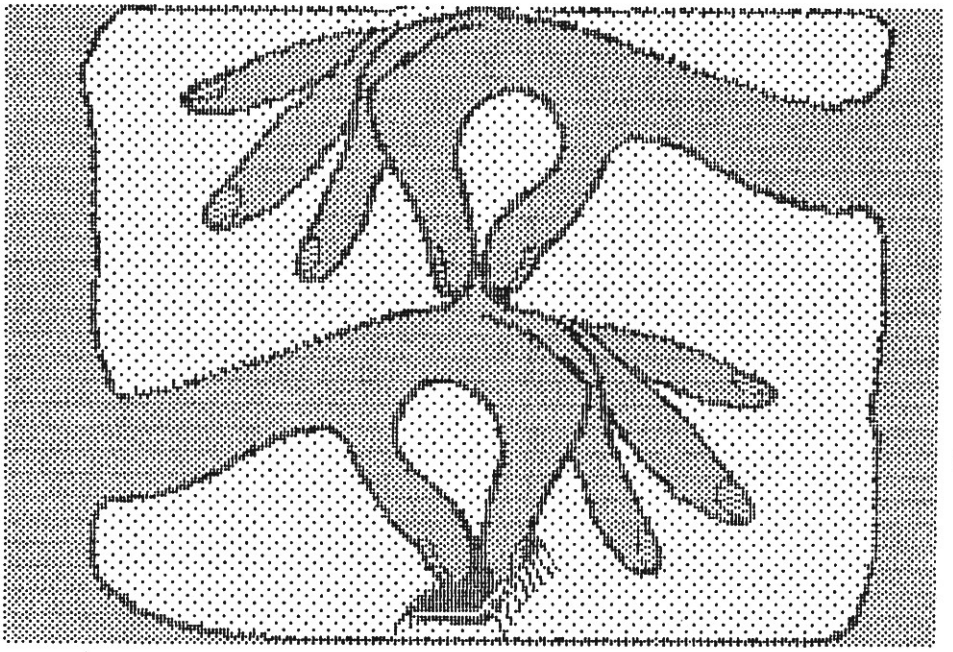
cím

Az információt, az adatokat azonban nem elég leírni és tárolni. Hiszen egy olyan gép, amelyik maga mindent tud, de ebből a gazdájával semmit sem közöl, nyilvánvalóan semmit sem ér. Az adatokat nem elég bevinni vagy kiszámolni a gépben, hozzájuk is kell férni, hogy tovább dolgozhassunk velük. Ezért minden adatot, amelyet a tárba visznek, külön *címen* helyeznek el. Ha minden egyes bitnek önálló, saját címe lenne, akkor ez a lehetséges címek számát borzasztó mértékben megnövelné. A cím maga sokkal hosszabb volna, mint az ott lévő adat. Ezért nem az egyes biteknek, hanem csoportjuknak, általában az egyes bájtoknak van önálló címe. S ha egy adat nem fér el egy bájtban, mert például több számjegyből vagy betűből áll (a legtöbbször ez fordul elő), azt több, egymást követő cím alatt helyezik el. Előkereséséhez megadják a legelső bájt címét és azt, hogy ezután még hány bájtot foglal el az adat.

Az első időkben a számítógépek tárolói jelfogókból épültek fel. Ezek legfeljebb néhány ezer bájtot tudtak tárolni, mégis szobányi nagyságú területet foglaltak el. A hatvanas években ezeket kiszorították a ferritgyűrűs tárak, melyekben apró mágnesek mágnesezettségi iránya jelezte, hogy az adott bitnek az aktuális értéke 0 vagy 1. Ezek tárolókapacitása már több millió bájtot is elért és néhány szekrényben fértek el. De nemcsak a térfogatuk volt nagy, hanem az energiafogyasztásuk is. Ezért a hetvenes években fokozatosan felváltották őket a félvezetős tárak, melyek igen nagy integráltságú áramkörökből épültek fel. Ezekről a következő fejezetben beszélünk kissé bővebben.

kilobájt

A gép nagyságának egyik jellemzője, hogy központi tárolójában hány bájt fér el. A tárolóméretet rendszerint a bájt ezerszeresében (pontosabban $2^{10} = 1024$-szeresében), *kilobájtban* (rövidítve kbájt), illetve milliószorosában ($2^{20} \approx 1$millió), megabájtban (rövidítve Mbájt) szokták megadni. A fejlődésre jellemző, hogy míg tíz évvel ezelőtt egy 512 kbájt tárolójú gép már nagy gépnek számított, addig ma már sok személyi számítógép memóriája is néhány ezer forint többletköltséggel minden nehézség nélkül kibővíthető erre a méretre, sőt a professzionális kategóriában a 640 kbájt a megszokott méretek közé tartozik.



Olyan szilíciumlapocskák, chipek, melyeken kb. 0.25 mm^2-en 256 kilobit tárolható, már sorozatban készülnek. Így a számítógépek memóriája, mely egy évtizeddel ezelőtt még hatalmas szekrényeket töltött meg, ma már gyufásdoboznyi helyen is elfér.

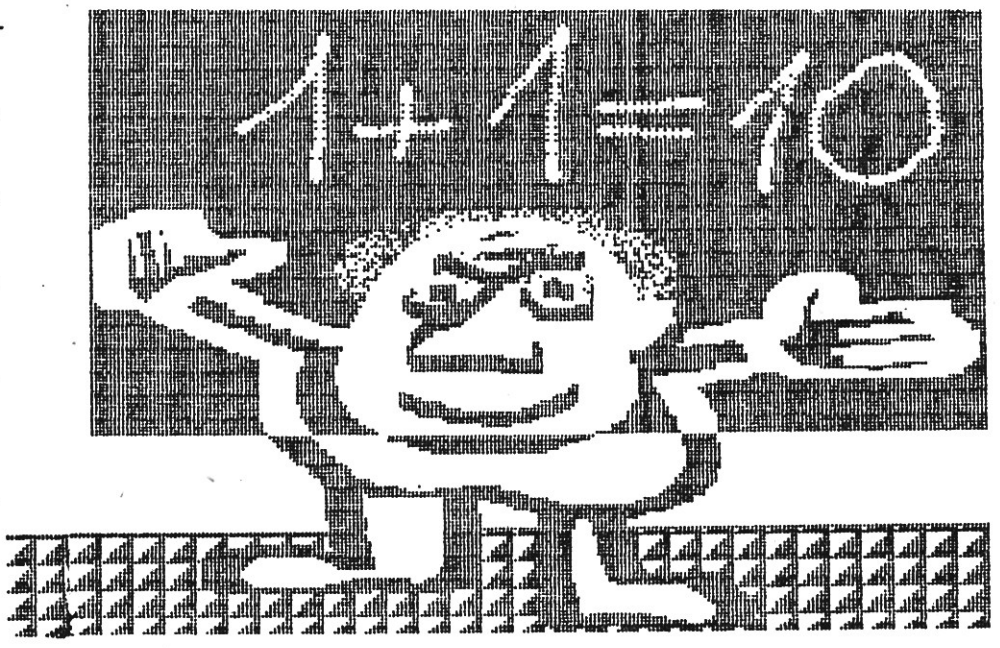
Chip-chip-chóka

A tár másik jellemzője a gyorsaság; mennyi idő alatt tud egy bájtot a címe alapján a gép az elvégzendő művelethez elővenni, illetve azt a művelet elvégzése után a helyére eltárolni. A korszerű professzionális személyi számítógépeknél, a kis- és nagyszámítógépeknél ez az időtartam az egymilliomod másodperc nagyságrendjébe esik, de vannak olyan speciális gépek, melyek még ennél is 10–100-szor gyorsabbak.

11.

A BUTA SZÁMÍTÓGÉP, AMELYIK KETTŐIG SEM SZÁMOL

Akár fejben vagy papíron, akár asztali számítógéppel, abakusszal vagy kézi kalkulátorral számolunk, percenként legföljebb néhány műveletet tudunk elvégezni. A műveletek elvégzéséhez szükséges időt az határozza meg, hogy milyen gyorsan gondolkodunk, írunk, billentyűzünk, illetőleg olvasunk. A korszerű elektronikus gépek a központi tárukban elhelyezett program alapján több százezer, sőt egy részük több millió műveletet is el tudnak végezni egyetlen másodperc alatt. Ez fantasztikus teljesítmény! A számítási műveletek mennyiségében bekövetkezett hatalmas növekedés következtében a számítógépek segítségével megoldható tudományos-technikai feladatokban nemcsak mennyiségi, hanem minőségi változás állt be. Ezzel kapcsolatban éppen azért mindenekelőtt egy igen általános tévhitet szeretnénk eloszlatni. A számítógépek legnagyobb részét ugyanis, nevükkel ellentétben, nem számítások elvégzésére, vagy legalábbis zömmel nem erre használják. Jelenleg a számítógépek többsége az üzleti, pénzügyi és kereskedelmi életben és az államigazgatásban üzemel, ahol a legfontosabb feladatuk a gyors és megbízható, állandóan naprakész információszolgáltatás. Ezek az információk különböző helyekről származó adatok összegyűjtése, rendszerezése, különféle szempontok szerint történő kiválogatása és csoportosítása révén állnak elő vagy közvetlenül az eredeti adatokból, vagy pedig rajtuk elvégzett matematikai és logikai műveletek eredményeként.



Ha azonban nem kápráztat el minket a számítógépek szédületes műveleti sebessége, és a dolgok legmélyére tekintünk, menten kiderül, hogy a legokosabb számítógép is nagyon buta. A mindennapi életben leginkább elterjedt digitális elektronikus számítógépek – a köznyelv általában ezeket nevezi számítógépeknek – csak az IGEN-t és a NEM-et tudják értelmezni és megkülönböztetni egymástól. Ez a gép belsejében azt jelenti, hogy egy adott áramkörön keresztül vagy folyik áram vagy nem, illetve a mágneses perifériákon azt, hogy egy két irányba mágnesezhető elem éppen melyik irányba van mágnesezve. Ezeket a fizikai jelenségeket aztán a gépnél sokkal bölcsebb matematikusok, akik tudták, hogy kettesben szép az élet, átfogalmazták a kettes számrendszerben való

számábrázolásra is, úgy, hogy a gép számára az IGEN (vagyis az, hogy egy áramkörben éppen folyik az áram) például az 1, a NEM (azaz, hogy nem folyik), a 0 számjegyet jelentik. A gép alapvető számítási ismeretei pedig mindössze a kettes számrendszerben történő alábbi egyszerű műveletekre terjednek ki:

$$\begin{array}{lll} 0+0=0 & & 0=0 \\ 0+1=1 & \text{továbbá:} & 0\neq 1 \\ 1+0=1 & & 1\neq 0 \\ 1+1=10 & & 1=1 \end{array}$$

A gép minden aritmetikai és logikai műveletet ezekre az alaplépésekre, valamint a számjegyeknek egy helyértékkel való jobbra vagy balra történő eltolására vezet vissza. Ezeket az egyszerű műveleteket aztán nagyon gyorsan csinálja, gyakran többet közülük egyidejűleg. Ráadásul ezeket a számításokat hibamentesen tudja elvégezni.

Ha például azt akarjuk, hogy a gép szorozzon össze két számot, akkor a szorzás műveletét jelentő jelsorozatot kell közölnünk vele. Ezt a hazánkban legjobban elterjedt ESZR (a szocialista országok egységes számítórendszerének rövidítése) valamint a világpiacot uraló IBM gyártmányú közép- és nagygépeknél megfelel a 00011100 számnak. (Valójában ez csak az egyik fajta szorzásnak a jele). Hasonlóan, az összes többi aritmetikai és logikai műveletnek, valamint minden egyéb utasításnak, amelyet a gép értelmezni tud megvan a saját, csak egyesekből és nullákból álló jele. Ezt az adott művelet *gépi kódjának* nevezzük. Természetesen nem minden géptípusnál egyforma a gépi kód. Lehetséges, hogy a fenti példában szereplő jelsorozat egy másik cég által gyártott számítógépnél szorzás helyett egy egészen más műveletet indít meg a gépben (vagy értelmezhetetlen butaságot jelent.)

gépi kód

Az első számítógépeknél a programokat is ilyen módon, utasításonként és azon belül számokként kellett leírni. Minden gépi utasításnak megfelelt egy programutasítás. Ezt hívták gépi kódban való programozásnak. Ez természetesen igen nehéz volt. Minden matematikai és logikai műveletet elemi lépésekre kellett bontani. Ezek mindegyikének nemcsak a numerikus kódjára kellett emlékezni, hanem arra is, hogy a műveletben szereplő operandusok az operatív tár milyen számú címein találhatók.

Ez a bonyolult programozási technika, a számítástechnika fejlődésével szerencsére igen hamar kiment a divatból. A gép hardverje nem lett okosabb, lényegében továbbra is ugyanúgy a fejezet elején leírt néhány alapműveletre vezet vissza minden feladatot, de az utasításokat már nem kell egyenként bináris formában megadnunk. Napjaink számítógépeihez már olyan programokat is mellékeltek, melyek képesek a matematikai képletekhez vagy az írott nyelvhez hasonló – bár meglehetősen szigorú szabályok szerint megfogalmazott – utasításokat átültetni a gép számára érthető gépi kódba. (Ezeket hívjuk fordítóprogramoknak, melyekről a 30. fejezetben még bőven fogunk szólni.) A holnap, vagy talán inkább a holnapután számítógépei pedig már a mindennapi élet nyelvén leírt, sőt elmondott utasításokat is meg fogják érteni és az azokban foglaltakat pontosan végre fogják hajtani.

A fordítóprogramok segítségével először sikerült elérni, hogy nem volt már buta számítógép, csak buta fordítóprogram. A szoftverfejlesztés technikájának javításával aztán sikerült oda is eljutni, hogy nincs már buta fordítóprogram sem, legföljebb buta programozó vagy buta rendszerszervező. Ez alól természetesen a könyv írói, szerkesztői, lektorai és legfőképpen figyelmes olvasói kivételek, akármilyen hosszadalmasan és hibásan készítik is el programjaikat.

12.

A HARDVER LEGFONTOSABB ELEMEI

A számítógép működési elvének megismerése után fordítsuk figyelmünket a vasalt részekre – akarjuk mondani a hardverre.

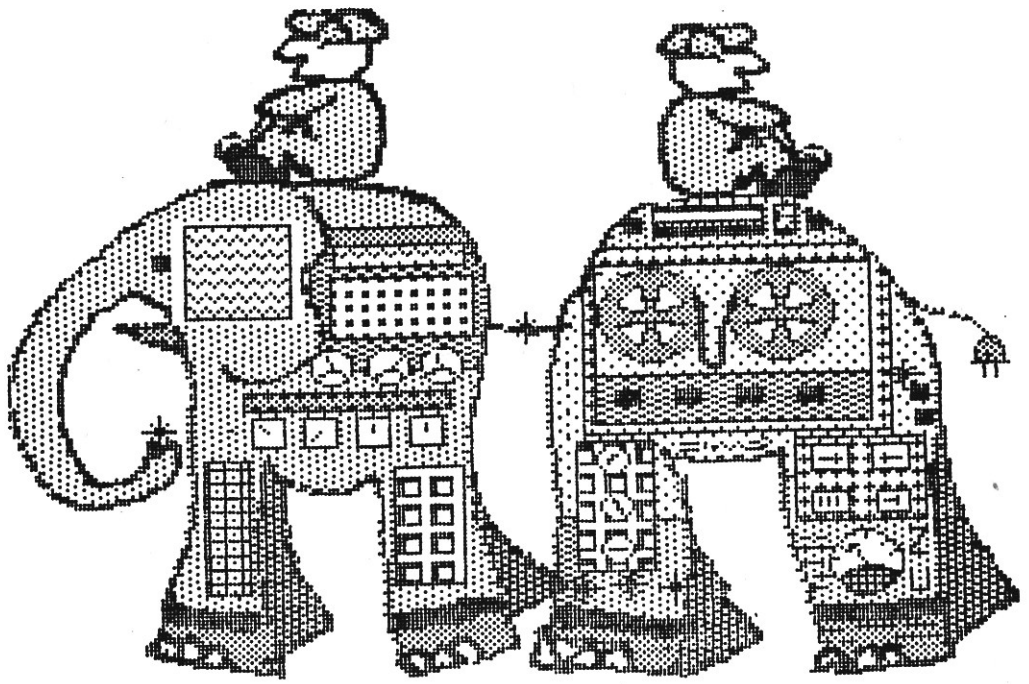
Egy számítógép kiépítettségét (idegen szóval konfigurációját) alapvetően befolyásolja a géppel megoldandó feladatok jellege. Vannak azonban olyan elemei a hardvernek, amely minden valamire való számítógépnél megtalálhatók. Ezek:

- a tár, a ROM és a RAM (lásd a 13. fejezetet!), amely tárolja a programutasításokat és a feladatmegoldáshoz szükséges adatokat,
- a vezérlő egység és az aritmetikai–logikai egység, amelyek segítségével történik a műveletek vezérlése és végrehajtása,
- a beviteli és kiviteli egységek, amelyek biztosítják a külvilággal való kommunikációt,
- a háttértárak, amelyeken a programok és az adatállományok megőrzése történik.

központi tár, perifériák

A számítógép *központi része* a *központi tár*, valamint a processzor. A processzor látja el a vezérlési és a műveletvégzési feladatokat.

A tárhoz kapcsolódnak a *perifériák,* amelyeket a számítógéppel való kommunikációt biztosító bemeneti-kimeneti egységek valamint a programok és adatok tárolására szolgáló háttértárak alkotják.



A központi tár

A központi tár *(operatív memória, memória)* feladata, hogy mindazokat az információkat tárolja, amelyek egy probléma megoldása során az adott pillanatban szükségesek.

operatív memória, memória

Így itt tárolódik az operációs- vagy más néven működtető rendszer (lásd a 31. fejezetet!), amely a számítógép erőforrásaival gazdálkodik és felügyeli a programok végrehajtását. A fennmaradó területen osztozkodnak a végrehajtandó programok. Minden program számára kijelölt terület két részre oszlik, az eljárásterületre, itt van a program, és az adatterületre.

(Ez az elválasztás valójában nem teljesen merev. A programutasítások között is lehetnek adatok elhelyezve, időnként kényszerből, mert nem elég nagy a rendelkezésre álló memória, vagy tévedésből – ami természetesen hibás programozásból adódik, s így helytelen eredményhez vezethet – felülírhatunk adatokkal utasításokat vagy fordítva, utasítással adatokat. A programok és adatok egymás mellett tárolásának elvét annak idején Neumann János fogalmazta meg.)

bájt, rekesz, cím, a rekesz tartalma

A központi tárban megvalósított tárolás alapegysége a *bájt.* Egy bájt nyolc bitet tartalmaz. Az egy bájtnyi információ elhelyezésére szolgáló tárhely a *rekesz.* A központi tár tehát a felhasználó szemszögéből rekeszekből épül fel. Minden rekesznek önálló *címe* van. A cím a rekesz azonosítására szolgáló bináris szám, amelyet tizenhatos számrendszerben szokás megadni. A címek fixen vannak a tárrekeszekhez rendelve. Egy konkrét cím tehát mindig ugyanazt a tárrekeszt azonosítja. A cím segítségével válik lehetővé a rekesz feltöltése és az ott tárolt érték visszanyerése. A rekeszekben elhelyezett információt a *rekesz tartalmának* tekintjük. A tartalom mindig egy bitsorozat, amely természetesen időről időre változhat.

A bájtból, mint a tárolás alapegységéből 2, 4, 8 bájtos fizikai tárolási egységek alakíthatók ki. Ezeket bizonyos (elsősorban nagygépes) rendszerekben rendre félszónak, szónak, duplaszónak nevezik. Általában azonban a bájtnál nagyobb tárolási egységek neve a mérettől függetlenül *szó.* Ezek kezdőcíme mindig a legalacsonyabb bájtjának címével azonos.

Amikor egy adatot el akarunk helyezni a központi tárban, három dolgot kell megadnunk: annak a rekesznek a kezdőcímét, ahonnan a tárolás kezdődjék, a hosszt, amely megadja hogy a tárolás hány rekeszben történjék, és magát a tárolni kívánt adatot.

szimbolikus címek, adat-regiszter, elérési idő, hozzáférési idő

A ma használatos programozási nyelvekben a tárban őrzött adatokra történő hivatkozás *szimbolikus címekkel* történik.

A központi tár a számítógép többi részéhez *regisztereken* keresztül kapcsolódik. A regiszter adatok és utasítások átmeneti tárolására szolgáló, igen gyors működésű áramkörökből felépített hardverelem. A számítógépben levő regiszterek egy része speciális célú, más részük a programozó által is használható, általános célú.

Két speciális, a tárhoz kapcsolódó regiszter a *címregiszter* és az *adatregiszter.* A címregiszterben van az a tárcím, amelyre a művelet vonatkozik. (Ez a művelet vagy a címre való beírás, vagy a címről való olvasás.) A művelet végrehajtását a tárvezérlő irányítja. Az adatregiszterből íródik be az információ a tárcímre, illetve ide kerül a kiolvasás eredményeképpen a cím tartalma.

Az az időtartam, amely alatt a tár egy beírási vagy egy kiolvasási műveletet végrehajt, a tár fontos jellemzője. Ezt *elérési* vagy *hozzáférési* időnek nevezzük. A félvezetős tárak megjelenése előtt a tárból való olvasás destruktív olvasás volt, ami azt jelentette, hogy az olvasás hatására az érintett rekeszekben lévő információ törlődött. Mivel így lehetetlen az adat ismételt használata, ezért a kiolvasott bitsorozatot vissza kellett állítani. Ehhez persze időre volt szükség. Amíg az érintett cím tartalma nem volt visszaállítva, a tárhoz nem lehetett ismételten hozzáfordulni (nem lehetett újabb tárba író, vagy onnan olvasó parancsot adni). A tárhoz való kétszeri hozzáférés közötti legrövidebb időt a tár *ciklusidejének* nevezzük. Destruktív táraknál a ciklusidő az elérési idő többszöröse is lehetett.

A számítógép központi tárával kapcsolatosan a felhasználó alapvetően két kérdést szokott feltenni:

- mekkora az adott tár kapacitása, vagyis egyidejűleg hány adat helyezhető el,
- mekkora az adott tár ciklusideje, vagyis mennyi idő szükséges egy adat tárolásához, illetve visszakereséséhez.

A tárkapacitás mérőszámairól a 10. fejezetben írtunk.

A számítógépek központi tára modulárisan építhető ki. Minden gép esetében létezik egy minimális tárméret, amelyet egy maximális méretig lehet növelni. A tárméret növelését a feladatok számának állandó növekedése indokolhatja. A legkisebb méretnek a működőképesség szab határt, a legnagyobbnak gazdasági és technikai megfontolások. A számítógépek központi tárára vonatkozó kapacitásadatok igen eltérőek lehetnek az alkalmazási területtől és a számítógéphez tartozó szoftvertől függően. Ma a nagyobb számítógépek központi tárkapcitása 4–16 Mbájt.

A félvezetős tárak megjelenése után már találkozhatunk olyan számítógépekkel, amelyeknek központi tára eléri, sőt meghaladja a 16 Mbájtot. Az otthoni mikroszámítógépekre a 64 kbájt méretű tár jellemző, a professzionális mikrogépekre a 256 vagy 640 kbájt.

ciklusidő

Vizsgáljuk most meg azt a kérdést, hogy mekkora a központi tár működési sebessége! Mint tudjuk, ezt a sebességet a ciklusidővel mérhetjük. A központi tár ciklusideje a különböző konkrét számítógéptípusoknál eltérő értéket mutat, mert alapvetően a gép belső felépítése és a megvalósítás technológiája határozza meg.

A félvezetős tárak megjelenésével a korábbi 1–2 mikroszekundum nagyságú ciklusidőt sikerült csökkenteni. Egy minden szempontból korszerűnek tekinthető számítógépnél 50–100 ns (nanoszekundum) nagyságrendű (1 ns = 10^{-9} s). Várható ennek az értéknek a további csökkenése, miután megjelentek olyan számítógépek, ahol a ciklusidő 30 ns alatti érték.

A tár valamennyi címére vonatkoztatva azonos a ciklusidő. A központi tárban levő adatok tehát függetlenül a táron belüli helyüktől ugyanannyi idő alatt érhetők el. A ciklusidő emberi fogalmaink szerint olyan kicsi, hogy jogosan mondhatjuk: az adat, amely a központi tárban van, „azonnal" hozzáférhető.

A processzor

A processzor feladata a gépben lévő program végrehajtásának előkészítése, a végrehajtás irányítása és az előírt műveletek elvégzése. E feladatokat a processzor akkor tudja elvégezni, ha képes:

- a szükséges tárolóterület címét meghatározni,
- adott cím tartalmát lekérdezni vagy oda adatot beírni,
- adatokon aritmetikai és logikai műveleteket végrehajtani,
- utasításokat előírt sorrendben kezelni.

Ezeket a funkciókat két önálló rész valósítja meg. A műveletvégrehajtást az *aritmetikai–logikai egység,* az előkészítést és irányítást a vezérlő egység végzi el. Az aritmetikai–logikai egység olyan hardverelemekből épül fel, amelyek dekódolják (megfejtik) a tárból kiolvasott utasítást, s az utasításban kijelölt műveletet az ugyanott megadott tár–címtartalmakkal végrehajtják.

A műveleteket az aritmetikai–logikai egység áramkörök segítségével valósítja meg. A különböző műveleteket különböző idő alatt képes megvalósítani. A végrehajtás ideje függ:

- a művelet operandusainak helyétől (regiszterben lévő operandussal gyorsabb ugyanaz a művelet, mint tárban lévővel),
- az adatábrázolástól (fixpontos operandusokkal gyorsabb a műveletvégzés, mint lebegőpontosokkal),
- az elvégzendő művelettől (összeadáshoz kevesebb idő kell, mint szorzáshoz). (Lásd még a 16. fejezetben!)

Az aritmetikai–logikai egység egy műveletet mikroszekundum (10^{-6} s) nagyságrendbe eső idő alatt végez el.

A műveletvégzés előkészítését és a műveletvégrehajtás irányítását a *vezérlőegység* végzi.

aritmetikai-logikai egység,

vezérlőegység,

gépi kód,

műveleti rész,

címrész

A vezérlőegység működésének legfontosabb feltétele egy utasításnyelv, amely a számítógép számára közvetlenül érthető módon (bináris formában) adja meg az elvégzendő feladat programját. Minden számítógépnek megvan a saját „anyanyelve", amelyet *gépi kódnak* nevezünk. Minden programnak a végrehajtáskor gépi kódúnak kell lennie.

Mielőtt a vezérlő egység egy gépi kódú utasítás végrehajtásához hozzákezd, négy információra van szüksége. Ezek:

- az elvégzendő művelet megnevezése (kódja),
- az első operandus címe,
- a második operandus címe,
- a megszülető eredmény leendő címe.

Ezenkívül még egy ötödik információ is szükséges, legkésőbb az utasítás végrehajtása után:

- a soronkövetkező utasítás tárbeli címe.

Ha minden utasítás ennyi információt tartalmazna, igen sok helyet foglalna el a program a központi tárban. Ezért az információk számát különböző módon csökkentik. Így például az eredmény általában valamelyik operandus címén keletkezik, a következő utasítás címének kijelölésére pedig egy külön regiszter szolgál.

Egy utasításnak így alapvetően kétféle információt kell hordoznia:

- meg kell határoznia, hogy mi az elvégzendő művelet,
- meg kell adnia, hogy hol található(k) a művelet operandusa(i).

Az elsőt az utasítás *műveleti része,* a másodikat az utasítás *címrésze* végzi. Ez utóbbi egy vagy két címet tartalmazhat. Az utasítások konkrét alakja számítógép típusonként eltérő, hasonlóan a többi hardver adottsághoz.

A hardver fogalomkörébe tartoznak a perifériák is, ezeket külön fejezetekben (19–25.) mutatjuk be.

13.

CSAK A SZÉPRE EMLÉKEZEM...
(ROM és RAM)

A kíváncsi embereket leggyakrabban az izgatja, hogy „Mit? Hogyan? Kivel?". Számítógépekkel kapcsolatban is ezek a leggyakrabban fölmerülő kérdések, legföljebb, – mivel még a robotok sem tekinthetők élő lénynek – a „Kivel" „Mivel"-re módosul. Ismertetéskor azonban az utolsó kérdés rendszerint háttérbe szorul. Mindenki tudja, hogy a gép mit csinál. Számol, mint ahogy a neve is mutatja. (Természetesen azért azt is tudnunk kell, hogy működése a számítási és logikai műveletek kombinációján alapul, a számoláson kívül adatokat, információkat dolgoz fel.) Hogyan számol? Gyorsan, pontosan, megbízhatóan. És mivel? Adatokkal! Az előző fejezetben arról írtunk röviden, hogy hol találjuk meg azokat az adatokat, amelyekkel a gép dolgozni tud.

Most ezeknek a tárolóknak a jellegzetességeiről írunk pár szót. Hogy vajon tartalmilag jók-e azok az adatok, azzal majd egy más alkalommal fogunk foglalkozni. A legtöbbször – legalábbis bizonyos határok között – ezt is tudja ellenőrizni a gép, sajnos nem mindig. Ha hibás adatok jönnek be, akkor rossz eredmények is mennek ki. „Szemét be, szemét ki", ahogy a számítástechnikusok mondják.

félvezető tárak

ROM

RAM

A korszerű számítógépekben igen nagy integráltságú áramkörökből felépült *félvezető tárakat* alkalmaznak. Ezek nagyon kevés áramot fogyasztanak, igen gyorsan és megbízhatóan működnek, mert nincsen bennük semmiféle mozgó alkatrész. A jelenleg használatos adattárolók közül a két legfontosabb típus a ROM és RAM. A *ROM* nem a memória állapotára utal, hanem az angol Read Only Memory (csak olvasható tároló) rövidítése. Ez olyan adatokat (leggyakrabban utasításokat) tartalmaz, melyeket a számítógép működése folyamán végig változatlan formában kell megőrizni. Ezért ezt a tárat a gép gyártója tölti fel, általában egyszer s mindenkorra. (Persze a számítástechnika is ismer kiskapukat: van olyan gép, melynek a ROM tárolója kicserélhető. E lehetőséggel olyankor élnek, ha a gépet egészen más jellegű feladatoknak a megoldására kívánják alkalmazni, mint amilyenekre azt eredetileg tervezték.)

A ROM tárolóelemei szilíciumlapkán kialakított diódák. E diódákat egy négyzetháló metszéspontjaiban helyezik el, s a háló mentén áramvezető réteggel kötik össze őket. Így bármelyik diódát két-két kivezetésen keresztül elérhetjük, s feszültséget – áramimpulzust – kapcsolhatunk rá. Az információ beírásakor bizonyos diódákra akkora áramot kapcsolnak, hogy az a diódát átégeti, s helyén a lapkán parányi lyuk keletkezik. Az információt – az azt leíró 0 és 1 számjegyet – a lapkán megmaradt diódák, illetőleg lyukak képviselhetik.

Ezek a tárak, minthogy beléjük égettük az információt, nem programozhatók újra. Ám készülnek olyan félvezető alapú tárak is, amelyeket a felhasználó maga programozhat egy különleges berendezéssel, s olyan is, amelynek tartalma törölhető, s a tár újra programozható. A PROM tárat (Programmable ROM, azaz: programozható csak olvasható tár) azoknak a felhasználóknak fejlesztették ki, akik egy bizonyos feladat megoldásához szükséges programokat, adatokat, információkat állandóan változatlan formában kívánják bent tartani a tárban. Ezek számára a speciális utasításcsoportot egyszer beégetik a PROM-

ba, és attól kezdve az állandóan benne lesz a tárban. Az EPROM (Eraseble Programmable ROM) törölhető és programozható ROM tár tartalma többször törölhető és újraprogramozható. Az ilyen tárak törlése ultraibolya fénnyel történik, de van elektromos úton is törölhető, ezt hívják EEPROM-nak.

A mikroszámítógépeknek a központi vezérlő egységében — ez hajtja végre a különféle számítási és logikai műveleteket, illetőleg vezérli az egész berendezés munkáját — többnyire csak állandó tartalmú ROM tárak vannak (a nagy gépekbe változtathatót is beépítenek). Egyetlen félvezető lapkán (chipen) több százezer bitnyi információ tárolható. Ha ennél nagyobb befogadó képességű tárra van szükség, több chipet helyeznek el egy nyomtatott áramköri kártyán.

A *RAM* az angol Random Access Memory (értelemszerűen fordítva: össze-vissza is elérhető tároló rövidítése. Ennek szokásos elnevezése: véletlen elérésű tároló, ami ugyan nem fedi a valóságot, hiszen rendszerint nem a véletlen dönti el, hogy melyik adatot akarjuk kiválasztani, hanem nagyon is tudatos szempontok.) Ebből nemcsak olvasni lehet az adatokat, hanem be is lehet őket írni, eltárolni későbbi felhasználásra. Ezeknél a technológiailag különféle módon felépülő memóriatípusoknál a legfontosabb összehasonlító szempontok a „Mennyit?", azaz hány adatot (bájtot) tudunk bennük tárolni, mennyi idő alatt tudunk elérni egy bájtot, és mennyibe kerül egy adat (bájt) tárolása.

Különleges szerepet tölt be az operatív tárban az a felhasználó által csak olvasásra elérhető tároló rész, amelyik a gép munkáját irányító operációs rendszer nélkülözhetetlen összetevőit tartalmazza. Ez fizikai felépítését tekintve RAM, speciális programozási eszközökkel átírható, de az alkalmazási programok készítői számára ROM-nak tűnik, hiszen csak olvasni tudnak belőle.

A számítógépek központi tára modulárisan építhető ki. Minden géptípushoz létezik egy minimális tárméret, melynél kisebbel nem üzemeltethető, és egy maximális nagyság, amelyig meghatározott lépésekben bővíthető. A kis méret növelését a feladatok számának és bonyolultságának növekedése indokolja. A tárméretnek technikai és gazdasági megfontolások szabnak határt.

Az igen nagy integráltságú és nagy integráltságú áramkörök (VLSI, LSI) igen gyorsak, de viszonylag drágák. Ezért ezeket csak a központi tárban használják. Az árban, hozzáférési időben és tárolókapacitásban jó kompromisszumot nyújtó mágneslemezekről, melyek önálló egységként a központi egységhez csatolhatók, a 22. és 23. fejezetben írunk majd. Ezek segítségével az otthoni személyi számítógépek 16–128 kbájtnyi tárkapacitása 160–720 kbájtra, a 256–2048 kbájt körüli professzionális személyi számítógépeké általában 10–80 Mbájtra, míg a néhány Mbájt operatív memóriájú nagy gépeké több száz, sőt ezer Mbájtra bővül ki.

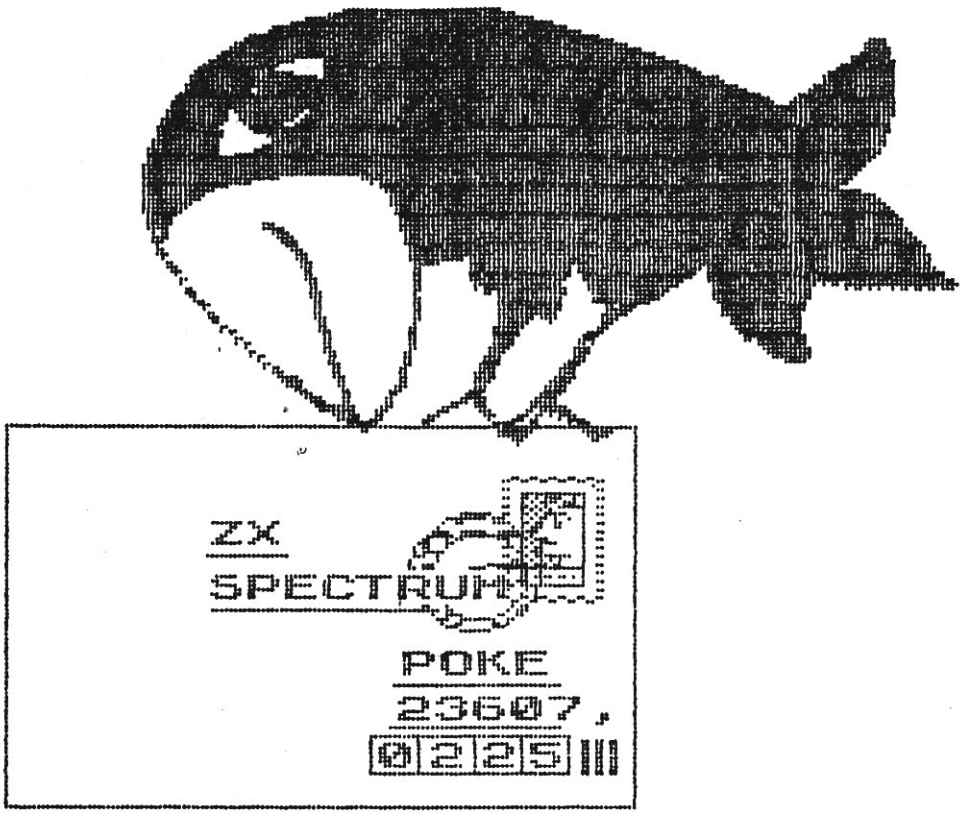
Természetesen az olcsóbb áron megvalósított tárolónövekedésért valamilyen módon fizetnünk kell. Ebben az esetben ez a gyorsaság. Az adatokat a mágneslemezen jóval hosszabb idő után találjuk csak meg, mint az operatív memóriában. Ezenkívül ahhoz, hogy az aritmetikai–logikai egység dolgozni tudjon velük, még be is kell őket olvasni a központi tárolóba. Természetesen ezek a gép műveleti sebességéhez viszonyított hosszú idők emberi szemmel nézve nem olyan borzasztóan nagyok, általában nem haladják meg a néhány század másodpercet. Ilyen módon a különböző tárolók ésszerű kombinációjával jó kompromisszumot érhetünk el a tárolható és feldolgozható adatok mennyisége és a feldolgozás gyorsasága között.

14.

MI A TEENDŐ?
(Utasítások)

Ahhoz, hogy a számítógép dolgozni tudjon, meg kell mondanunk számára, hogy mit csináljon és mivel. Erre szolgálnak a program utasításai, melyeket a gép a saját nyelvére lefordítva, ún. gépi kód formájában tárol a központi memóriában és lépésenként hajt végre. (Kivéve az interpreter típusú nyelveket.) Minden mennyiségnek, adatnak amelyekkel a gép dolgozni tud, meghatározott helye, mint mondtuk címe van a központi tárban. Ez megállapodás alapján az adatot tartalmazó bájtsorozat legelső bájtjának a címe. A bájtsorozatokra a program készítésekor címük alapján hivatkozhatunk, és a gépi kódban megadott utasításban már bináris formában szerepelnek. Az utasítás végrehajtásakor a gép mindig az éppen azon a címen (vagy címeken) lévő adattal (vagy adatokkal) hajtja végre az utasításban meghatározott műveletet.

Címzésminta



Az *utasításnak,* mely maga is bináris jelsorozatként, és rendszerint 2, 4 vagy 6 bájtban tárolódik, két fő része van. Az első az *utasításkód,* amely megmondja, milyen műveletet kell végrehajtani (pl. szorzás, kivonás). Ezenkívül, ha ugyanazon adatokat különböző formában is lehet tárolni (pl. számokat bináris, decimális, vagy lebegőpontos alakban, lásd a 16. fejezetet), akkor azt is megmondja, hogy a tényezők helyén lévő bitsorozatot milyen tárolási módban kell értelmezni. Az utasítás második része meghatározza a műveletben résztvevő tényezők és az eredmény *címét* és, ha szükséges, a hosszát is (pl. szorzandók és szorzat címe, az összehasonlítandó jelsorozatok címe és hossza). Ahhoz például, hogy a

utasítás, utasításkód, cím

szocialista országok egységes számítógépes rendszerében (ESZR) gyártott gépek két adatot összehasonlítsanak, egy 6 bájtból álló bitsorozatot kell utasításként megkapjanak. Ebből az első bájt az utasítás (itt összehasonlítás) gépi nyelven megadott kódját, esetünkben 11100101-et kell tartalmazzon. A második bájt azt határozza meg, hogy a 3. és 4., illetve az 5. és 6. bájtban megadott címektől kezdődően hány bájtot kell összehasonlítani. Ha mondjuk az összehasonlítandó bájtok száma kettő és az első címen

1 1 1 1 0 0 0 0 1 0 1 0 0 0 1 1

van, míg a másodikon

1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1,

akkor az összehasonlítás eredménye „nem egyenlő", mert az aláhúzott helyen levő bit az első összehasonlítandóban 0, a másodikban 1. Ennek a „nem egyenlő" jelzésnek a hatására a gép logikai döntést tud hozni és például az összehasonlító utasítást közvetlenül soron követő címről veszi a következő utasítást, míg „egyenlő" jelzés esetén egy másik helyen tárolt utasítással folytatná. Így például, ha személyi száma alapján keresünk egy személyt, akkor ha a szám nem egyezik meg a keresettel, tovább folytatjuk a keresést.

Ha minden egyes utasítást ilyen módon kellene megadnunk, tudnunk kellene minden utasításnak a bináris kódját és operandusai bináris tárcímét. Ez esetben a programok megírása rettenetesen hosszú időt venne igénybe. Szerencsére erre ma már nincs szükség. Lehetőség van ezeknek az utasításoknak az ember által könnyen megtanulható és alkalmazható nyelven való megadására, melyeket aztán a gép maga fordít le saját gépi kódjára. Ezekről a lehetőségekről majd később, a programozási nyelveknél és fordítóprogramoknál fogunk részletesebben beszélni. (Lásd a 32. fejezetet!)

Az utasítások száma, a gép utasításkészlete és fajtája típusonként változó. Megvalósíthatósági módjuk is különbözhet. A drágább, gyorsabb gépeknél egyes összetett utasításokat önálló, erre a célra specializált hardver valósít meg. Az olcsóbb gépeknél csak a legfontosabb alaputasításokat végzi el közvetlenül a hardver. A többieket a gépbe egyszer s mindenkorra bevitt mikroprogramok (a ROM-ba beépített utasítássorozat) bontják le a tényleges hardvernek megfelelő elemi utasításokra. Ezzel természetesen a programozónak nem kell törődnie, az ő számára ezek a mikroprogrammal végrehajtott utasítások ugyanúgy egyetlen egy gépi utasításnak tűnnek, mint hogy ha azt a hardver közvetlenül hajtaná végre. Különbséget legföljebb abban észlel, hogy a mikroprogrammal végrehajtott utasítások jóval hosszabb ideig tartanak. A korszerű gépek utasításkészlete 50–200 különböző utasításból áll, melyeknek jelentős része – a legdrágább nagy gépek kivételével – mikroprogramokon keresztül valósul meg.

Az utasítások a következő fő csoportokba sorolhatók:

aritmetikai utasítások

Aritmetikai utasítások. Ezek tényleges számolási műveleteket hajtanak végre az operandusokon. Általában a négy alapművelet valamelyikének elvégzésére szolgálnak. Különböző formáik vannak attól függően, hogy a művelet tényezői bináris, decimális vagy lebegőpontos számok (lásd a 16. fejezetet!). Ebbe a csoportba sorolhatjuk az egyik számábrázolási módból a másikba, például binárisból decimálisba, átalakító műveleteket is.

Az aritmetikai műveleteket minden gép a leggyorsabban a belső lelkivilágához legközelebb álló bináris számokon végzi. Két bináris szám összeadása egy nagy, de azért még nem szupernagy és szupergyors gépen 0,1–1 milliomod másodpercet vesz igénybe. Ugyanaz a művelet ugyanazon a gépen lebegőpontos számokon mintegy háromszor, decimális formában kb. tízszer több időt vesz igénybe, mint bináris ábrázolásban. A műveleti sebességekben észlelhető eltérés oka az, hogy a bináris aritmetikán működő

áramkörök jóval egyszerűbbek, és így gyorsabbak, mint a decimális és lebegőpontos aritmetikát megvalósító áramkörök.

logikai utasítások, vezérlésátadó (ugró) utasítások, kitüntetett utasítások

Az utasítások következő csoportjába a *logikai utasítások* tartoznak. Ezek az adatok összehasonlítására, más tárolóhelyre való átvitelére, logikai feltételek kiértékelésére szolgálnak.

A *vezérlésátadó (ugró) utasítások* teszik lehetővé a program menetének a feldolgozás pillanatnyi állapotától, részeredményeitől függő megváltoztatását. Normál körülmények között az utasítások a tárban való elhelyezkedési sorrendjüknek megfelelően egymás után hajtódnak végre. Ez általában megfelel a programutasítások leírási sorrendjének. Olyan nyelvekben azonban (pl. Basic), ahol mindegyik utasításnak sorszáma van, ez nem a megírási, hanem a számozási sorrendnek felel meg, mivel az értelmező/fordító program először e szerint a sorszám szerint sorba rakja az utasításokat. Bizonyos esetekben azonban ettől mindenképpen (feltétlen ugró utasítás) vagy valamilyen feltételtől függően (feltételes ugró utasítás) el kell térni. Az ugróutasítás végrehajtása után a következő utasítást a vezérlőmű az ugróutasításban megadott új címtől kezdi, és innen ismét sorban folytatja a végrehajtást egészen addig, amíg egy újabb ugróutasításhoz, vagy a program végéhez nem ér.

Vannak olyan utasítások, amelyek – különösen akkor, ha a gépen egyszerre több program is fut – hibás használat esetén az egész gép működését félbeszakítják, mert más programokat, vagy értékes adatokat tehetnének tönkre. Ezért ezeket a felhasználók által írt szokványos programokból nem lehet közvetlenül kiadni. Végrehajtásukat csak kérni lehet az egész gép működését ellenőrző, irányító felügyelő programtól, amely előbb gondosan megvizsgálja, hogy a kért utasítás problémamentesen elvégezhető-e, és ha igen, akkor maga gondoskodik a végrehajtásáról. Ezeket a különleges körültekintéssel kezelendő utasításokat *kitüntetett utasításoknak* nevezzük. Ide tartoznak például a számítógép ki- és bemeneti egységeivel kapcsolatot létesítő utasítások.

15.

KI KERÜLHET AZ FTC ELÉ?
(Betűk, számok, jelek ábrázolása)

Mint azt már többször említettük, a mai számítógépek munkájának csak egy részét teszik ki a tényleges számolási műveletek. Tevékenységük jelentős részét nevekből, szavakból, betűkből álló adatokkal végzik el.

Ahhoz azonban, hogy a számítógépek ne csak számoljanak, hanem szavakat, elnevezéseket írjanak ki, azonosítsanak, névsorokat rendezzenek ABC-sorrendbe, szükséges, hogy ezekben a gépekben ne csak számokat tudjunk ábrázolni, hanem betűket, betűkből és számokból összeálló szavakat, kifejezéseket is. Ugyanakkor még a legbonyolultabb számítógépek alapáramkörei is csak a 0 és az 1 között tudnak különbséget tenni, csak ezzel a két jellel tudnak dolgozni a kettes számrendszerben.

Ennek látszólag ellentmond, hogy a legtöbb számítógépnek a bináris aritmetika mellett van lebegőpontos aritmetikája is, sőt a drágábbaknak decimális is, aminek segítségével közvetlenül képesek számolási műveleteket végrehajtani lebegőpontos alakú illetve tízes számrendszerbeli számokon is. Valójában azonban ezeket a műveleteket az utasítás kiadásával egyidejűleg a számoló egység felbontja meghatározott bináris műveletek sorozatára, ezt hajtja végre, majd az így kapott eredményt visszaalakítja a lebegőpontos ill. decimális számoknak megfelelő bináris ábrázolási formába.

Azt, hogy egy adott címen levő bitsorozatot bináris, lebegőpontos vagy decimális számként kell értelmezni, az aritmetikai művelet kódja határozza meg. Más a szorzás utasításának a kódja, ha a tényezők bináris számként vannak tárolva, más, ha decimális és megint más, ha legebőpontos számként találhatók a memóriában. Ugyanez érvényes a többi matematikai műveletre is.

EBCDIC kód

A nagygépeknél nemzetközileg általánosan elfogadott kódrendszerben (szokásos elnevezése az angol meghatározás rövidítése: *EBCDIC kód*) 1 bájtban, azaz 8 bitben ábrázolunk egy jelet. Az ABC nagy betűinek a kódja A = 11000001-től Z = 11101001-ig terjed, éspedig úgy, hogy az ABC-ben előrébb levő betűnek megfelelő bináris szám értéke kisebb. Betűknek, szavaknak az összehasonlítását tehát a gép bináris számok összehasonlításával végzi el, és az ABC-sorrendbe való rendezés bináris számoknak nagyság szerinti sorbaállításának felel meg.

Így ha arra vagyunk kíváncsiak, hogy neve alapján a számítógép az FTC-t vagy az MTK-t helyezi előbbre, akkor a három betűből, három bájtból álló név az FTC-nél az F = 11000110, míg az MTK-nál az M = 11010100 bitsorozattal kezdődik. Ebből már látjuk, hogy az első szám lesz a kisebb. A gép az FTC-t egyértelműen az MTK elé helyezi. Ha olvasóink közül ezzel többen nem értenének egyet, úgy ne a gépet hibáztassák. „Ő" jól válaszolt. Legföljebb mi emberek nem tettük fel helyesen a kérdést, hogy ezt az eredmény kaptuk. Ha a rangsorolást nem a név, hanem a pályán a legutóbbi időkben nyújtott teljesítmény alapján (pl. az utolsó fordulókban szerzett pontok, a Népsport osztályozókönyve, a bajnokágban elért helyezés) kértük volna, akkor lehet, hogy más eredményt kaptunk volna.

A személyi számítógépek többsége a betűket, számsorokat, írásjeleket egy másik szabvány szerint ábrázolja. Ezt *ASCII kódnak* nevezzük. Egy betűt, számot, speciális jelet itt is egy bájtban helyeznek el, de itt a nagybetűs angol ABC karaktereinek kódja 01000001-től 01011010-ig terjed. A 00000000–00011111 karakterek (0–31) speciális utasításokat jelentenek. Így például a 00000111 érték (7) észlelésére a legtöbb típusú személyi számítógép hangjelzést ad, a 00001100 (decimális 12) letörli a teljes képernyőt, a 00001101 (13) pedig a következő új sor elejére állítja a kurzort, és lehetővé teszi új karaktereknek a gépbe való beírását. (A BASIC programozási nyelvben a PRINT CHR$ (7), PRINT CHR$ (12) illetve a PRINT CHR$ (13) utasítás hatása.)

ASCII kód

Mivel azonban a betűk bináris kódjának a növekedése megfelel az ABC sorrendjének, az FTC-t a személyi számítógépek is az MTK elé sorolják. Akinek az nem tetszik, és továbbra is csak elnevezésük alapján szeretné az MTK-t jobbnak látni az FTC-nél, az tegye fel a számítógépnek a nagy kérdést: „Melyik a nagyobb csapat, az MTK vagy az FTC?" Mivel az M betű kódsorozata az EBCDIC kódban is és az ASCII kódban is nagyobb bináris szám, mint az F betűé (ASCII-ben 77 illetve 70, ami megfelel a bináris 01001101 illetve 01000110 számoknak) a nagy- és kisszámítógépek is egyaránt az MTK-t tartják nagyobbnak.

Összehasonlítva ezt azzal, hogy ugyanaz a gép az előző kérdésnél viszont az FTC-t mégis az MTK elé helyezi, milyen tanulságot vonhatunk le?

a) Egy fecske nem csinál nyarat, de egy Verebes már igen. (Legalábbis a kézirat leadásának időpontjáig!)
b) A számítógép nem ért a focihoz és össze-vissza zagyválja a dolgokat.
c) Ahhoz, hogy a számítógép értelmes válaszokat adjon, értelmesen kell az adatokat közölni vele és értelmesen kell a kérdéseket feltenni.

Ha az olvasó az a) vagy b) választ tartja helyesnek, akkor tegye le a könyvet és menjen el a legközelebbi FTC–MTK örökrangadóra. A c) válasz elfogadása esetén folytassa tovább számítástechnikai ismeretei bővítését, még viheti valamire ezen a területen is, – de ettől függetlenül azért szintén megtekintheti a mérkőzést. Lehet, hogy azt is érdekesnek fogja találni.

16.

NEMCSAK KETTESBEN SZÉP AZ ÉLET
(Számábrázolás)

Már többször hangsúlyoztuk, hogy a számítógép közvetlenül csak a bináris számjegyeket tudja értelmezni, csak ezekkel tud dolgozni. A mindennapi életben viszont tizes számrendszerben számolunk. Ezen kívül szükségünk lehet olyan nagy és olyan kis számokra is, melyek egységes ábrázolása a kettes számrendszerben száznál is több számjegyet igényelne. Nem beszélve az előjelről, hiszen az aztán egyáltalán nem közömbös, hogy az egyenleg a bankszámlánkon plusz vagy mínusz 5000 Ft.

Bár a tízes és bináris számrendszerek közötti állandó átváltás, de az igen sok helyi értékű bináris számok ábrázolása sem okozna problémát, megvalósításuk nagyon gazdaságtalan volna. Ezért a legtöbb számítógép az aritmetikai műveleteket háromféle számábrázolásban tudja elvégezni. Azt, hogy az adott címen elhelyezkedő bináris számsorozatot hogyan kell értelmezni, a művelet kódja szabja meg. Más például a szorzás műveletének a gépi kódja, ha a tényezők bináris, decimális vagy lebegőpontos számok.

decimális ábrázolás, bináris számábrázolás, lebegőpont

A *bináris számábrázolásban* a számok 2 vagy 4 bájton helyezkednek el. A legelső bit értéke 0, ha a mögötte álló szám pozitív és 1, ha negatív. A fennmaradó 15 illetve 31 bit a szám abszolut értékét adja meg binárisan. A bináris számok mindig egész számok. Ilyen módon a legnagyobb szám, amivel a gép dolgozni tud $2^{31}-1 = 2\ 147\ 483\ 647$, ami decimális ábrázolásban 9 jegyű számoknak felel meg. (A tízjegyű decimális számoknak csak egy része fér el.) Így például az 1987-es számot a 2 bájtos, ún. rövid bináris formában a 0000011111000011 bitsorozat adja meg.

Decimális ábrázolásban egy decimális számjegy 4 biten fér el. Itt az utolsó fél bájt a szám előjele, a decimális szám többi számjegyét pedig az utolsó bájt első fele, illetve a megelőző bájtok (ezek kettesével) jelölik. Nem az egész szám, hanem az egyes helyi értékeken álló számjegyek külön-külön vannak átszámítva kettes számrendszerbe.

Ebben az ábrázolásban az 1987-es számot a

0000	0001	1001	1000	0111	1111
	1	9	8	7	+

jelsorozat határozza meg. (Az egyes félbájtok közé az áttekinthetőség érdekében egy elválasztó vonalat helyeztünk el, és az ábrázolt decimális jegyet aláírtunk. A számítógépben ezek természetesen nincsenek.)

Decimális ábrázolásban a legtöbb gép maximum 15 jegyű számokkal tud dolgozni. Ezek lehetnek egész vagy tizedes számok is. Magát a műveletet azonban az aritmetikai egység egésznek tekintett számokon végzi el.

Sokszor azonban a decimális aritmetika lehetőségei sem elegendőek, pl. dolgozni kell 15 jegynél hosszabb számokkal is. Erre szolgál a lebegőpontos számábrázolás. Ez 4 vagy 8 bájton történhet.

A *lebegőpontos* számábrázolás a logaritmusnál tanult tízes számrendszerbeli normál számábrázolásnak a 16-os számrendszerbeli megfelelője. A legelső bit itt is a szám előjelét jelenti. Hogy ne kelljen negatív számokkal dolgoznunk a karakterisztikában, ezért az

előjel bitet követő 7 bit a 16-nak nem azt a hatványát jelenti, amellyel az utána következő bináris számot (bináris pont a szám előtt) meg kell szoroznunk, hogy az ábrázolt értéket kapjuk meg, hanem annál 64-gyel többet. (Így a 65-ös karakterisztika $16^{65-64} = 16^{1}$-vel, a 62-es $16^{62-64} = 16^{-2}$-vel való szorzást jelent.) A legnagyobb és legkisebb ábrázolható szám abszolut értéke 16^{63} illetve 16^{-64} (mintegy 10^{76} és 10^{-77}), a legtöbb értékes számjegy pedig 7, illetve a 8 bájtos hosszú formában 16 jegy lehet. Speciális gépekkel és/vagy programokkal ennél nagyobb pontosság is elérhető.

A korábbi példában már bemutatott $1987 = 7 \times 16^{2} + 12 \times 16^{1} + 3 \times 16^{0}$ lebegőpontos ábrázolásban tehát a következőképpen néz ki:

01000011	01111100	00110000	00000000
+ 67	7 12	3	
e kar.	mantissza*		

(ahol e = előjel, kar. = karakterisztika)

A mantissza $7 \times 16^{-1} + 12 \times 16^{-2} + 3 \times 16^{-3}$.

Leggyorsabban a gép, mint azt már az utasításokról szóló fejezetben írtuk, a saját „lelkivilágához" legközelebb álló bináris számokkal dolgozik.

De még a lényegesen lassúbb lebegőpontos aritmetikai egység teljesítménye is jó személyi számítógépeknél másodpercenként 10—100 ezer, nagygépeknél 0,1—1 millió, szuper teljesítményű gépeknél pedig 100—500 millió művelet körül van.

---

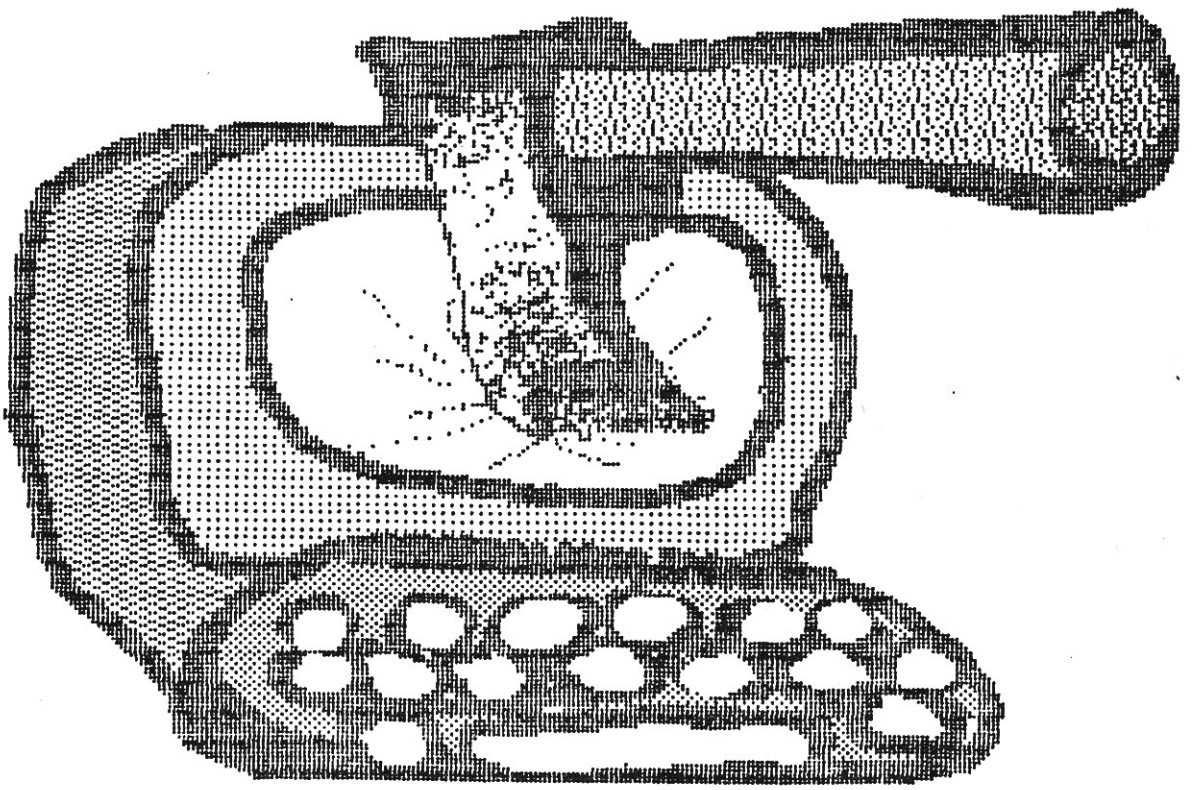
*A mantissza itt mást jelent, mint a matematikában! (Lektor megjegyzése.)

17.

A TÉVEDHETETLENSÉG TITKA

(Hardverhibák elleni védelem)

Gyakran halljuk, olvassuk, hogy ezt vagy azt az eredményt számítógép szolgáltatta. Ennek közlésével a riporter, az újságíró az eredmény hitelességét, pontosságát kívánja hangsúlyozni. Magunk is többször hivatkoztunk arra, hogy a számítógép nem tud tévedni, nem hibázik. Ennek ellenére mégis hallhatunk számítógépek által elkövetett óriási baklövésekről. Az egyik legismeretesebb ezek közül az volt, amikor az egyik légitársaságnál a munkabeosztást készítő számítógép értesített egy utaskísérőt, hogy a következő hónaptól kezdve kizárólag földi szolgálatra osztja be. Az indokolás: a szabályzat szerint az alkalmazottak terhességük negyedik hónapjától nem dolgozhatnak fedélzeten. A dologban csak egy bökkenő volt, – az értesítés az egyik férfi utaskísérőnek szólt.



Hardverhiba

Egy másik esetnél a számítógép az egyik alkalmazottnak a bankszámlájára egyszerre csak a korábbi járandóságának tízszeresét kezdte átutalni. A revizorok a tévedést csak fél év múlva derítették fel. Amikor megkérdezték az illetőt, miért nem szólt, mindössze ennyit válaszolt: „Azt hittem, végre rájöttek a cégnél, hogy valójában mennyit ér a munkám."

Az ilyesfajta számítógépes történeteknek se szeri, se száma. Egy részük valóban megtörtént, más részük könnyen megtörténhetett volna.

Hogyan egyeztethetők össze ezek a tények a számítógépek hibátlan működésében való hitünkkel?

Valójában ezen alkalmakkor nem a számítógép hibázott, hanem az ember. A programozó, aki rossz programot írt, vagy az adatrögzítő aki hibás adatot vitt be. Ha a gép rossz megbízást, hibás adatot kapott és annak ez lett a vége, arról nem tehet. Ő – a maga szemszögéből és technikailag nézve is – hibátlanul végezte el a rábízott feladatot.

Természetesen a legdrágább, legkorszerűbb, legjobb számítógép sem tökéletes. Áramkörei és méginkább mechanikus részei meghibásodhatnak. Még ha csupán 10 milliárd műveletből tévedne egyszer a gép, akkor is, mivel másodpercenként kb. 1 millió műveletet végez el, átlagosan 10 ezer másodpercenként, azaz mintegy 3 óránként előfordulhatna egy hiba. Ennek következményeit, azt hogy ez egy fontos számolási eredmény első, legértékesebb vagy utolsó, valójában érdektelen számjegye lesz-e, nem lehet előre tudni.

Az ötvenes években és a hatvanas évek elején készült gépeknél még valóban ilyen gyakorisággal fordultak elő meghibásodások. A mai, integrált áramkörökből felépített gépeknél azonban ennél jóval hosszabb a két meghibásodás között átlagosan eltelt idő. Ennek ellenére a számítógép áramkörei minden műveletet, minden adatátvitelt még külön is ellenőriznek. Erre szolgál a *paritásbit*. Ezt minden 8 bitből álló bájthoz a hardver írja hozzá automatikusan úgy, hogy értéke 1 lesz, ha a bájtban levő 1-esek száma páros és 0, ha páratlan. Ezáltal a paritásbittel kiegészített bájtban az egyesek száma mindig páratlan. Ha most egy bit értékét tévesen vinné át a gép, azaz pl. 0 helyett 1-et olvasnánk be, akkor ebben a bájtban (a paritásbitet is hozzátéve) az 1-ek száma páratlanról párosra változik. De az 1-ek számának páratlanságát, a paritást a hardver minden tárműveletnél bájtonként automatikusan ellenőrzi. Ha nem jó, akkor azonnal megszakítja a programot és paritáshibát jelez. Hogy ugyanabban a bájtban egyidejűleg két bit hibásodjon meg és így ezt a paritásbit ne jelezze (az 1-ek száma vagy nem változott, vagy 2-vel nőtt, illetve csökkent, így továbbra is páratlan maradt), annak valószínűsége a mai nagy megbízhatóságú áramköröknél gyakorlatilag zérus. (Ugyanígy dolgozhatnánk páros paritással is, amikor az egyesek száma egy bájton belül mindig páros számúra egészül ki. Bizonyos – elsősorban adatátviteli – rendszereknél be is lehet állítani, hogy a berendezés páros, vagy páratlan paritással dolgozzék.)

paritásbit

Ha az adatátvitel nem a központi tárban történik, hanem külső tárolóról vagy tárolóra, ahol a mechanikus mozgás és az adathordozó felületére esetleg rárakódó apró szennyeződések miatt lényegesen nagyobb a hiba valószínűsége, egyéb, automatikus hardver-hibaellenőrzés is van. Ezek közül a legfontosabb az, hogy az egy kiíró/beolvasó műveletben átvitt bájtokat a hardver kiegészíti általában két *ellenőrző bájttal,* melyeknek értékét az átvitt bájtok tartalma alapján számítja ki. Ha az átvitelben hiba volt, akkor a művelet végén újra kiszámolt ellenőrző bájtok nem fognak megegyezni az eredetiekkel. Ez az ellenőrzés igen sok esetben nemcsak az egyszeres, hanem a többszörös átviteli hibákat is kijelzi. Bizonyos esetekben még azt is megmutatja, hogy melyik bit volt hibás, s így azt automatikusan ki is tudja javítani.

ellenőrző bájt

Gyakorlati tapasztalat, hogy az elektronikus berendezésekben a legtöbb hiba a készülék bekapcsolásakor jelentkezik. Tovább csökkenti annak a valószínűségét, hogy ne már a folyamatos üzemelés közben vegyük észre a hibát az a lehetőség, hogy a bekapcsolás után módunk van külön erre a célra szolgáló ellenőrző programokkal megvizsgálni a berendezés minden egyes részét. Drágább személyi számítógépeknél és nagy számítógépeknél ezeknek a vizsgálatoknak egy része (pl. az operatív memória minden egyes bájtjának az ellenőrzése) bekapcsoláskor automatikusan megtörténik. Más részüket (pl. a mágneslemezek bájtonkénti megvizsgálását, a nyomtató minden karakterének ellenőrzését) a felhasználók vagy a számítóközpont karbantartó technikusai, mérnökei végzik el az erre a célra szolgáló hibavizsgáló programok lefuttatásával.

18.

(NE) JÁTSSZUNK A KÁRTYÁKKAL!
(A személyi számítógép architektúrája)

A kártya szóról mindenkinek tök meg makk jut eszébe, kivéve azokat akiknek pikk meg treff. De ha egy hardver szakember meghallja ezt a szót, úgy illik, hogy neki más is eszébe jusson. Kártyának nevezik a műszakiak azokat a paneleket, amelyeken a számítógép különböző funkcióit megvalósító hardver elemei helyezkednek el. A hardverelemek méretcsökkenésével a panelek mérete is zsugorodott, sőt kialakultak bizonyos „célszerű" elrendezések.

Tapasztalatból tudjuk, hogy a személyi számítógépek küllemre nagyon különböznek egymástól. Ez elsősorban az otthoni számítógépek kategóriájában figyelhető meg. Érdekes módon a professzionális kategóriába tartozó gépek legtöbbje, ha talán esztétika és formatervezettség tekintetében nem, de lényegét tekintve nagyon hasonló felépítésű. (Ennek oka, hogy majdnem mindenki az első igazán sikeres professzionális gép, az IBM cég XT és AT névre hallgató gépeinek felépítését utánozza. Így azután ezek az utánzatok nemcsak az eredetire, hanem egymásra is hasonlítanak.)

Nézzünk meg egy ilyen gépet közelebbről.

alapkártya, többfunkciós kártya

A gép legfontosabb részei egy házikószerű dobozban vannak elhelyezve. Némi gúnnyal azt mondhatjuk, hogy efféle házikót mindenki tud gyártani. Ha felnyitunk egy ilyet és belepillantunk, paneleket, ún. kártyákat találunk egymás mellett. Legfontosabb közülük az *alapkártya.* Ezen helyezkedik el maga a processzor, amely a gép vezérlője, aritmetikai–logikai egysége. Itt találhatók azok az áramkörök (IC-k) is, amelyek a RAM tárat alkotják. Ezeknek az áramköri elemeknek a száma nemcsak attól függ, hogy maga a gép mekkora RAM tárral rendelkezik, hanem sokkal inkább attól, hogy egy ilyen elemben mekkora tárkapacitást tudnak a gyártók megvalósítani. Sok gépben találunk 64 kbites egységeket. Ezekből az egységekből 8-at, illetve 16-ot kötnek párhuzamosan egy bájt illetve szó befogadására képes rekeszek illetve regiszterek kialakítása céljából. Ma már a 256 kbites elemek a kereskedelmi forgalomban sem számítanak nagynak, a japánok 2 dollár alá(!) szorították ennek az árát. A processzoron és a RAM-on kívül az alapkártyán van elhelyezve a billentyűzet-illesztő és jónéhány segédáramköri elem.

Egy másik kártya biztosítja a képernyő vezérlését. Ennek a kártyának a felépítésétől függ többek között, hogy az adott gép színes-, vagy monochrom (egyszínű) képernyővel tud-e dolgozni. A hajlékonylemezek (floppy-k) vezérlését általában külön kártya biztosítja. Egy ilyen vezérlőkártya kettő vagy négy egységet képes kezelni. A winchester, vagyis a fixlemezes tár vezérlője vagy ugyanezen a kártyán vagy külön kártyán található.

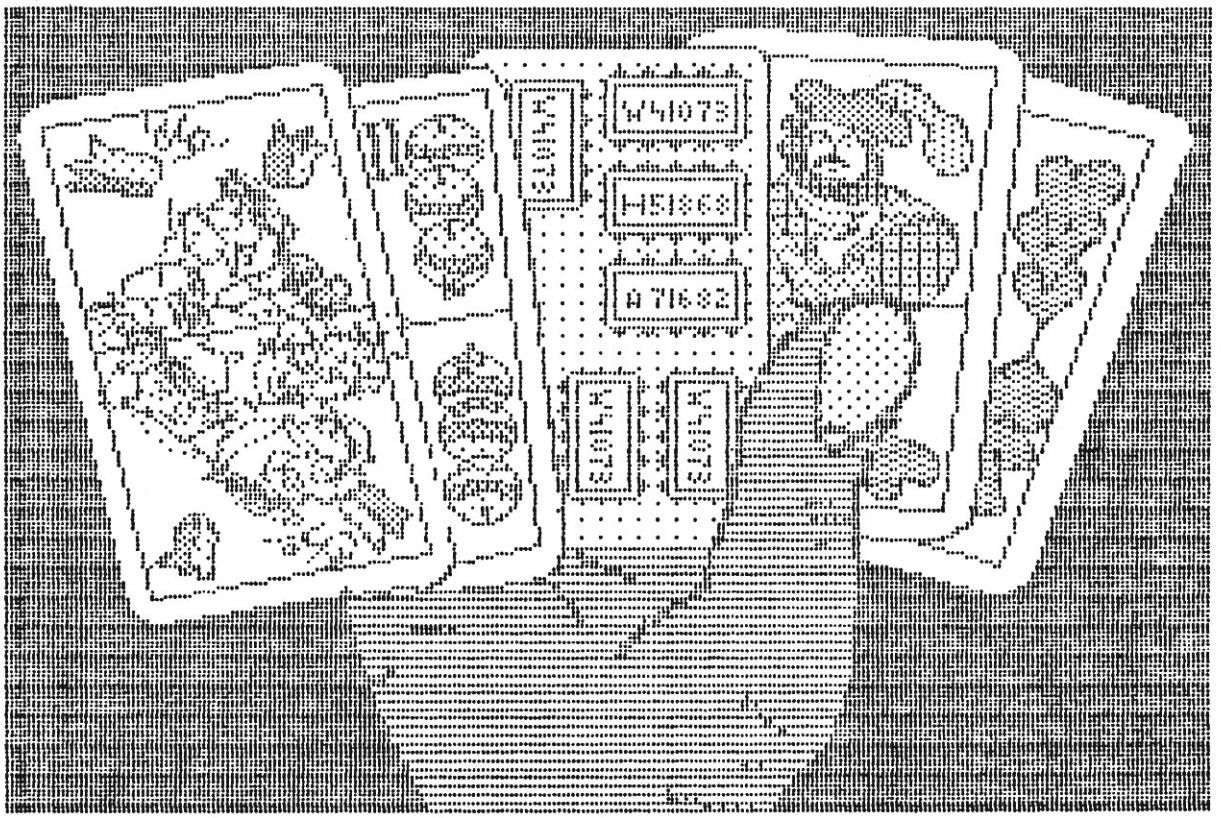
A *többfunkciós kártyán* kapnak helyet a perifériák (pl. nyomtató) illesztőáramkörei, az óra és más segédáramkörök. A gép sebességét alapvetően az óra frekvenciája határozza meg. Az órajel frekvenciáját megahertzben (MHz) mérik. 1 MHz órajel mellett a gép műveletvégzési ideje 1 mikroszekundum v. μs. Ez kis teljesítménynek számít. 5 és 10 MHz közötti órajel már jó gépre utal, de a csúcsteljesítmény jelenleg a 100 MHz környékén van. E házikóban található még a gép tápegysége, amely az energiaellátást biztosítja.

A gépben külön helyet szánnak a hajlékony lemez(ek) meghajtója és a fixlemezes tár számára. A professzionális kategóriában többnyire a két hajlékony lemezes meghajtó vagy az egy hajlékony lemez + egy fixlemez a leggyakoribb összeállítás. A fixlemez számára ugyanakkora hely kell, mint a hajlékony lemezes meghajtó számára.

monitor, billentyűzet

A *monitor* (képernyő) és a *billentyűzet* mindegyike külön egység. Ezek kábellal csatlakoznak a gép többi részéhez. Szokásos, de csak kényelmi szempontból, hogy a monitort a gép tetejére helyezik el. Ez természetesen másképp is lehet. Ugyancsak praktikus szempontok miatt a billentyűzetet a házikó elé helyezik, hiszen így lehet kényelmesen látni a képernyőt, kezelni a lemezeket stb.

Önálló egység a nyomtató (ha van egyáltalában), amely szabványos csatlakozóval illeszthető a számítógép (általában) hátoldali részéhez.



szinkronizálás

Az eddig elmondottakból kitűnik, hogy a processzorhoz többféle, eltérő tulajdonságú és működésű periféria csatlakoztatható. Vajon hogyan tud a számítógép ennyi fajta perifériát kiszolgálni? A válasz meglepő: nem veszi észre, hogy eltérő fizikai tulajdonságú egységekkel áll szemben. Ennek egyik fő oka a csatlakozás módjában van. Csatlakozási felületnek (idegen szóval interfésznek) nevezzük azokat a hardverelemeket, amelyek a különböző berendezések és eszközök egymáshoz való illesztését megvalósítják. A csatlakozási felületnek nemcsak a mechanikai szintű illeszkedést kell biztosítaniok, hanem a logikai és adatkapcsolati szintet is. A mechanikai szint azt jelenti, hogy a két csatlakozófelület összeilleszthető. A logikai szintű egyezőség körébe tartozik, hogy az átvitt információ típusa, kódolása, az átvitel időzítése megegyezik a felület két oldalán.

Az adatátvitelen kívül biztosítani kell vezérlő jelek átvitelét is, amelyek a két oldal működésének szinkronizálását végzik. A csatlakozási felületeknek nemzetközi szabványokban meghatározott módon kell viselkedniök. Ez biztosítja, hogy pl. egy bárhol gyártott géphez, bármely más országban gyártott periféria csatlakoztatható legyen. Mindez nagyon bonyolultnak tűnik, sőt az is. Megvalósításuk viszont rengeteg kényelemmel jár.

19.

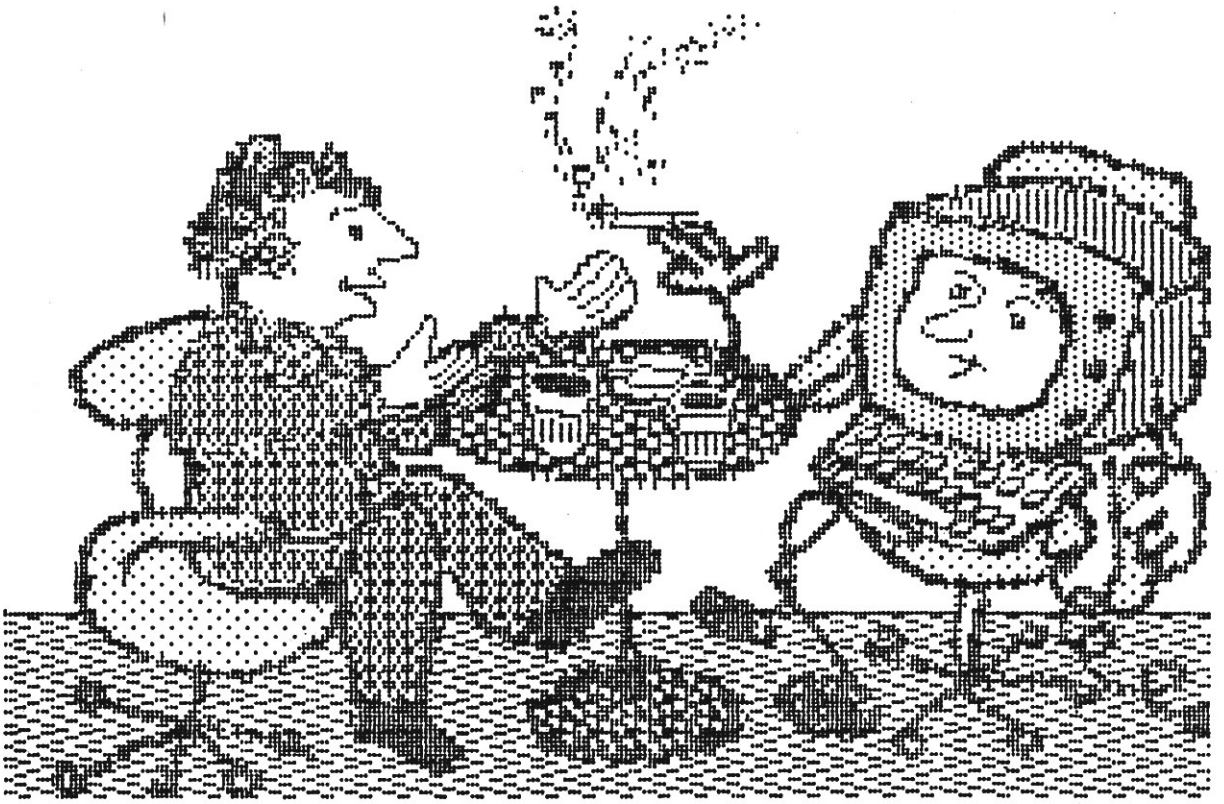
PÁRBESZÉD A GÉPPEL
(Adatmegjelenítők)

Aki manapság leül egy terminál, vagy egy személyi számítógép elé és barátságosan „elbeszélget" a géppel, talán el sem tudja képzelni, hogy még egy évtizeddel ezelőtt is az volt a szokásos eljárás, hogy a számítógépbe beviendő adatokat, programokat először valamilyen, a gép által is olvasható adathordozóra (lyukszalagra, lyukkártyára, később mágnesszalagra vagy mágneslemezre) rögzítették. Az így összegyűjtött adatokat azután egy kötegben olvastatták be a gépbe. Az egyes programok az eredményeket – természetesen egymástól elkülönítve – egy (nagyszámítógépeknél esetleg több) nyomtatóberendezésen nyomtatták ki papírra (így keletkezett a tabló), míg az üzemeltetéshez szükséges üzenetek egy speciális készüléken, a *konzol-írógépen* jelentek meg. Az adatbevitel és ellenőrzése ezen a módon meglehetősen nehézkes volt. Az esetleges hibák javítása csak külön lépésben, a korábban bevitt adatoknak a gépi ellenőrzés utáni kiíratásával volt lehetséges. Ráadásul a hibák a gépnél derültek ki, míg az aki ezeket javíthatta, valahol máshol dolgozott. Azok az eredmények, amelyekről első ránézésre azonnal megállapítható lett volna, hogy teljesen használhatatlanok, nagy tablókra, majd viharos sebességgel a szemétgyűjtőbe kerültek, egyszerűen azért, mert nem volt mód a „ránézésre". (Ilyen tablókból ragasztott zacskókkal találkozhatunk néha a piacokon, ha gyümölcsöt vásárolunk.)

konzol-írógép

Jelentős mértékben megjavult az ember és a gép közötti kapcsolat az *adatmegjelenítő* berendezések, másnéven *képernyős terminálok* vagy adatvégállomások elterjedésével.

képernyős terminálok



A számítógéphez közvetlenül, vagy adatátviteli vonalon keresztül csatolt terminál egy adatbeviteli célú billentyűzetből és egy olvasást, vizuális megjelenítést nyújtó képernyőből áll.

billentyűzet

A *billentyűzeten* leütött jel azonnal megjelenik a képernyőn, de a számítógépbe csak egy speciális billentyű lenyomásának hatására jut be az adatbevitel során. Ha az adatrögzítő észreveszi, hogy valamelyik adat hibás, akkor azt még a gépbe való beküldés előtt kijavíthatja a helyes értékre való átírással. Lehetőség van arra is, hogy a beküldött adatokat a gép formailag és logikailag is ellenőrizze, és ha hibát talál bennük, akkor azonnal jelezze és kérje kijavításukat. (Ehhez persze jól megírt felvivő-ellenőrző program kell.)

Ugyancsak fontos eszköze a terminál a programfejlesztésnek, hiszen így a programozó gyorsan tudja programjait tesztelni, javítani. Közte és a számítógép között nincs ott az ügyfél- és diszpécserszolgálat az összes bürokratikus nehézségeivel.

A jelek és számok megjelenítésére használt képernyőkön egy sorban rendszerint 80 karakter (betű, szám, vagy más írásjel) és 24 sor fér el. Ha megtelt a kép, akkor újabb sor írásakor az egész képernyő tartalma följebb tolódik egy sorral. A legfelső, a legrégebben leírt sor eltűnik és az újonnan bevitt sor kerül legalulra. Amikor a gép ír ki a képernyőre, lehetőség van arra is, hogy a kiíratást leállítsuk, ne sorokat, hanem tejes képernyőoldalakat írassunk ki, de csak akkor, ha az előzőt már elolvastuk.

A személyi számítógépek megjelenítői a terminálok képernyőjéhez hasonlóan viselkednek. Az otthoni gépekhez kapcsolt közönséges TV-készüléken általában csak 40 karakter fér el egy sorban, igaz a betűméretük sokkal nagyobb.

A képernyőn különféle ábrákat, grafikonokat, rajzokat is megjeleníthetünk. Különösen hasznos ez mérnöki tervező munkában, vagy vezetői döntések hatásának előrejelzésénél, de a számítógépes játékoknál is.

Az igazi párbeszéd, mint neve is mutatja, két fél szóváltása. Ma már vannak számítógépek amelyek tudnak beszélni — szintetikusan előállított emberi hangot generálva. Az emberi hang digitalizálása (s így a számítógépben való tárolása) is megoldott probléma. Így nincs messze az idő, amikor már nemcsak a gyerek, hanem a számítógép is visszafelesel, ha véletlenül valamilyen feladat elvégzésére utasítják.

20.

AZ ÍRÁS MEGMARAD
(Érintéses nyomtatók)

Az ember azért találta ki az írásjeleket, hogy gondolatait rögzíthesse. Az írás segítségével képes az ember tudását átadni a mindenkori utókornak is. Ez az igény a számítógépek megjelenésével sem változott meg, sőt tovább nőtt, hiszen a gép egyik feladata éppen az információszolgáltatás. Ráadásul az ember telhetetlen is, mert jól olvasható szép betűkre vágyik.

A számítógép papírra író egységei, a *nyomtatók* nehezen felelnek meg ennek az igénynek, mert amelyik nyomtató gyorsan képes írni, az általában nem ír szépen, amelyik meg szépen ír az lassú. És mindegyik drága, mert nagyon sok mechanikus alkatrész van bennük. (Persze ez a lassúság még mindig gyorsabb, mint egy gépírókisasszony több órás átlagteljesítménye, de már hozzászokhatott az olvasó, hogy a számítógépektől más sebességeket és szolgáltatásokat várunk el, mint mondjuk a titkárnőktől.)

Nagyon sokféle nyomató van forgalomban, mindenki megkeresheti a céljának megfelelőt. A vezérigazgató titkárnője kaphat levélminőséggel, de lassan író berendezést. A nagy tömegű feldolgozást végző számítóközpont pedig gyors, de nem igényes nyomtatókat.

Valamiféle rendet teremteni a nyomtatók között

- a nyomatatás módja,
- a nyomtatás sebessége,
- a nyomtatott kép minősége

alapján lehet. A nyomtatás módja szerint két nagy csoportot különböztetünk meg:

- az érintéses nyomtatókat, melyeknél a nyomtatott kép előállításakor a papírhoz az írófej hozzáér,
- a nem érintéses nyomtatókat, ahol a papír érintése nélkül jön létre a nyomtatási kép.

Ebben a fejezetben az érintéses nyomtatókat mutatjuk be, az érintés nélküli nyomtatókkal a következő fejezetben fogunk foglalkozni.

A hagyományos számítógépek esetében, ahol a felhasználó nem is találkozik közvetlenül a géppel, a számítógép által előállított információkhoz történő hozzájutás egyik fontos formája a nyomtatott eredmény, amelyet a szakmában *tablónak* neveznek. A papírra nyomtatás legnagyobb problémája ez esetben a megfelelő írási sebesség biztosítása. Az teljesen nyilvánvaló, hogy a hagyományos karakterenkénti nyomtatás, amely percenként néhány száz írásjelnél nem tud többet megjeleníteni, nem megoldás. A nagyszámítógépek érintéses nyomtatói nem jelenként hanem soronként nyomtatnak. Az ilyen típusú berendezéseket *sornyomtatóknak* nevezik.

tabló, sornyomtatók

A sornyomtatók két leggyakrabban használt változata a hengeres és a láncos nyomtató. Mindkettőben a kiírandó sor minden betűhelye mögött egy kis kalapács van. Ám a hengeres változatban a henger egy-egy alkotója mentén ugyanaz a jel van, s ez annyiszor ismétlődik, ahány hely összesen van a kiírandó sorban. Nyomtatáskor a betűhenger folyamatosan forog a papírhengerrel párhuzamos tengely körül. Amikor a betűhenger megfelelő

jele a papír fölé kerül, a kalapács – egy festékszalagon keresztül – ráüt a papírra, s a jel képe megjelenik a papíron. Egyidejűleg a sor minden azonos jeléhez tartozó kalapács működik. Így legkésőbb a betűhenger egyszeri körülfordulási ideje alatt elkészül egy-egy sor. A láncos nyomtatóban a jelek egy zárt láncon helyezkednek el, s a lánc a nyomtatandó sorral párhuzamosan mozog. E megoldásban a nyomtatókalapácsok akkor ütnek a láncra, amikor a kinyomtatandó betű a sor megfelelő helye fölé érkezik. Egy jó minőségű, megbízható, nagysebességű sornyomtató ára 1–2 millió forint.

A henger forgási, a lánc mozgási sebességének a mechanika korlátot szab. Ha kevesebb jellel megelégszünk (pl. csak a nagybetűk, számok, néhány írásjel), akkor ugyanaz a jel a hengeren több alkotón, a láncon többször helyezhető el. Így ha a jelek kétszer háromszor ismétlődnek, akkor nem egy körülfordulás kell a sor kiírásához, hanem csak annak fele, harmada. Ezért a sornyomtatók betűkészletének terjedelme fordított arányban van a nyomtatási sebességgel, vagyis bővebb betűkészlettel (kisbetűk használatával) lassúbb a nyomtatás. Ezért általában cserélni szokták azt az „alkatrészt" amelyen a betűk képe található. Így a különleges betűket nem követelő szövegek (pl. programok, számszerű eredmények) a kisebb jelkészletű eszközökön gyorsabban nyomtathatók. Így ahol a sebesség a kulturált szöveg iránti igénynél is nagyobb, a nyomtató nem tartalmazza például az összes magyar betűt.

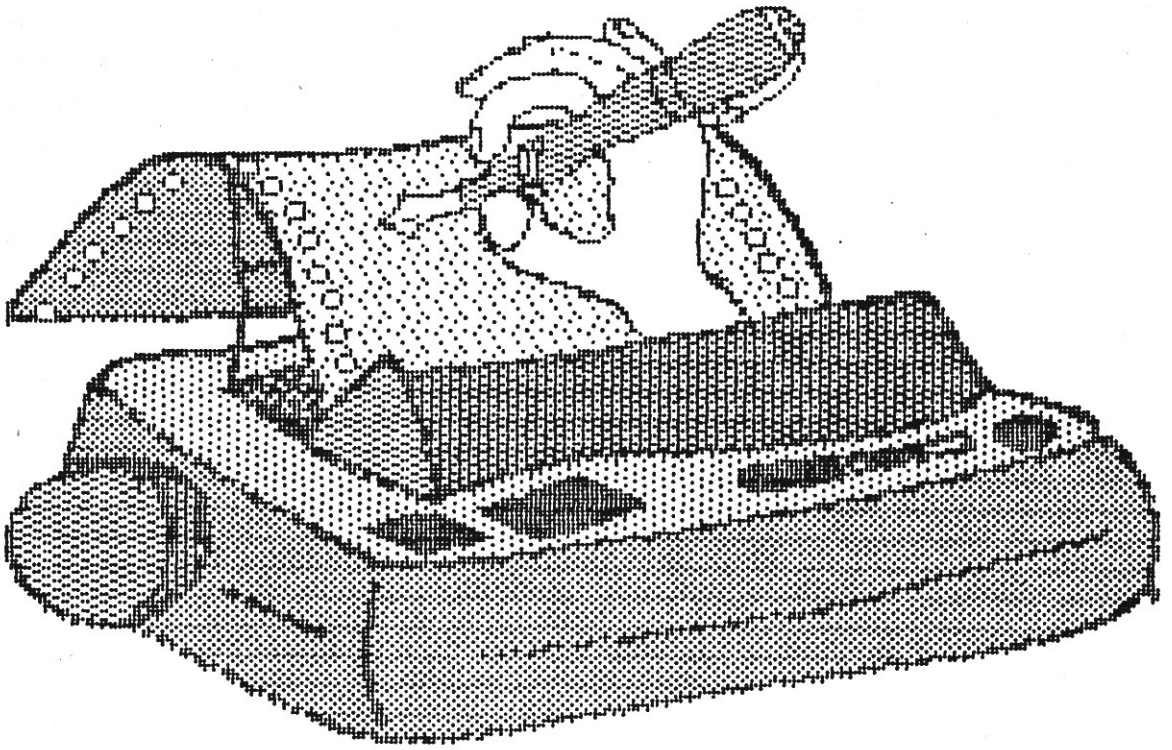
A sornyomtatók sorszélessége 120, 132 vagy 160 karakter. Egy-egy lapra maximálisan 60–72 sor nyomtatható.

mátrix-nyomtatók

A professzionális személyi számítógépek megjelenésével a nyomtatók iránti igény itt is megjelent, de jelentkezett a hordozhatóság (kis súly, kis térfogat) igénye is. A személyi számítógépekhez kapcsolható, a fenti kívánalmaknak megfelelő nyomtatóberendezések egyik csoportját képezik a *mátrixnyomtatók.*

Nevüket a nyomtatandó jel *kialakításának módjából* kapták. Ezeknél a nyomtatóknál a nyomtatott jel nem *úgy jelenik meg, hogy a nyomtatandó jel fém képe* ***valamilyen festéket tartalmazó anyagon keresztül a*** papírra nyomódik. Ez esetben a jelek pontok sorozatából állnak össze. Általában a jelek magassága 9 pontnyi, szélességük 7 pontnyi. Egy ilyen 9x7 táblázatban (a matematikában az ilyet nevezik *mátrixnak) a létező, illetve a hiányzó pontok* alakítják ki a nyomtatási képet. A mátrixnyomtató tehát karakterenként állítja elő a szöveget. A gépben lévő karaktergenerátor lehetővé teszi a kisbetűk és a magyar hosszú magánhangzók nyomtatását is. A grafikus *mátrixnyomtatók segítségével gyakorlatilag bármilyen jel* ***nyomtatása lehetséges. Így grafikai jelek (e könyv ábrái) vagy*** akár japán, kínai írásjelek is előállíthatók a megfelelő programmal.
(Pl. ∡ β ∀ ∂ ∈ △ Γ Σ μ π √ ♪ ± ← ↓ ↑ → x^2 H_2O $^1/_2$)

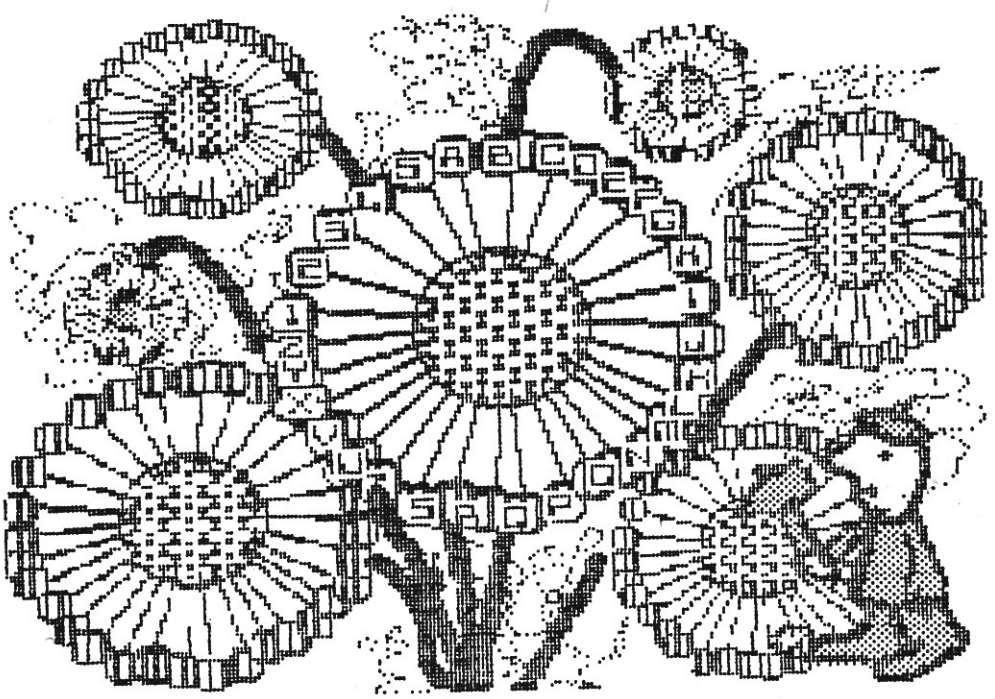
Az írásminőség és a nyomtató ára is összefügg. Bonyolultabb mechanikával elérhető sokkal finomabb mátrix (például 24 x 24) alkalmazása is. Az ilyen nyomtató írásképe eléri az írógépét. Ára viszont messze magasabb.



margaréta-kerekes nyomtatók

A levélminőség igénye hozta létre a *margarétakerekes nyomtatókat.* Ezeknél egy sugárirányban sűrűn bevágott műanyag tárcsa szélének a mentén helyezkednek el a karakterek képei. Minden karakter a tárcsára merőleges irányba képes kissé elmozdulni. A tárcsa forog, s amikor a kívánt karakter kerül nyomtatási pozícióba, egy kalapács-szerű szerkezet a tárcsa szelvényét a papírhoz nyomja. A nyomtatott kép a festékszalag segítségével kerül a papírra. Nevét onnan kapta, hogy egy-egy ilyen tárcsa a margaréta virágjára hasonlít: a szelvények a virág egy-egy szirmára emlékeztetnek. A különböző karakterkészletű tárcsák cserélhetők. Sebességük kisebb, grafikát nem tudnak, zajszintjük magas. Ez az ára a kifogástalan nyomtatási képnek.

De ha a címzett közel áll a szívünkhöz, nekünk sem mindegy, hogy amit írtunk, hogy néz ki.



21.

NE TAPOGASS!
(Nem érintéses és színes nyomtatók)

Az ember telhetetlen. Korábban örült annak, hogy sikerült elfogadható sebességű vagy megfelelő minőségű nyomtatókhoz hozzájutnia, meghozva a szükséges áldozatot a sebesség–szépség–ár hármasprobléma oltárán. Mostanában pedig igénye támadt rugalmas grafikai lehetőségekkel ellátott, csendesen dolgozó, (még gyorsabban, még szebben író), s újabban még színeket is tudó nyomtatók iránt.

És a fejlesztők igyekeznek kielégíteni ezeket az igényeket is!

lézer-nyomtató

Nagyobb sebességű, grafikát is egyszerűbben biztosító, csendesen működő nyomtató elfogadható áron: ez a *lézernyomtató.*

A lézernyomtató egyik legfontosabb eleme egy dob, amelynek felülete fény hatására homogén töltésűvé válik. Ezt a dobfelületet pásztázza végig egy lézersugár, hasonlatosan ahhoz, ahogy a tv képernyőt az elektronsugár. A lézersugár hatására a dobfelület homogén töltése megváltozik: ahol a sugár a felületet éri, ott a töltés kisül. A sugarat a nyomtatandó szövegnek illetve ábrának megfelelően lehet vezérelni. A dobfelületen kialakított töltéskép a dobbal érintkező papírra viszik át. A papír elhalad a fémszemcsékkel kevert, festékporos állomás előtt. Az elektromos töltéssel ellátott festékpor a papír töltetlen pontjaira tapad rá. Ezután már csak arról kell gondoskodni, hogy ott is maradjon. Ez a festékrészecskék „sütésével" történik meg. Ha mindez megvan, s közben a dob homogén töltöttsége is visszaállt, kezdődhet a következő lap nyomtatása. Ugyanezen eljáráson alapul a xeroxrendszerű fénymásoló.

A lézernyomtatók tehát egy egész lapot nyomtatnak egyszerre. Képminőségük vetekszik a margarétakerekes nyomtatók képével, problémamentesen tudnak grafikát készíteni. Csendesen működnek, amit az értékel csak igazán, aki dolgozott már folyamatosan működő sornyomtató, vagy mátrixnyomtató mellett.

A lézernyomtatók asztali változatainak sebessége kb. 8 lap/perc, s egy ilyen berendezés ára 2000 dollár körüli (s nyilván egyre csökken). A számítóközpontok számára készített igen nagy teljesítményű lézernyomtatók sebessége elérheti a percenkénti több száz oldalt – igaz, hogy áruk a több százezer dollárt. Ilyen sebesség mellett már nem is használható a klasszikus leporelló. Általában folyamatos papírra dolgoznak, s a szétvágás, válogatás a nyomtatás után történik.

Ha valaki lézernyomtatót akar saját számítógépe mellé beszerezni, csak akkor tegye, ha az üzemeltetéséhez szükséges pénzzel is rendelkezik. A festékkazetta és a dob néhány ezer oldal után cserére szorul, ez mintegy száz dollárnyi költség. Ha azzal vigasztalódnék, hogy ezt a párezer oldalt csak nagyon sokára érné el – akkor ne is vegyen lézernyomtatót, mert nem lesz kihasználva.

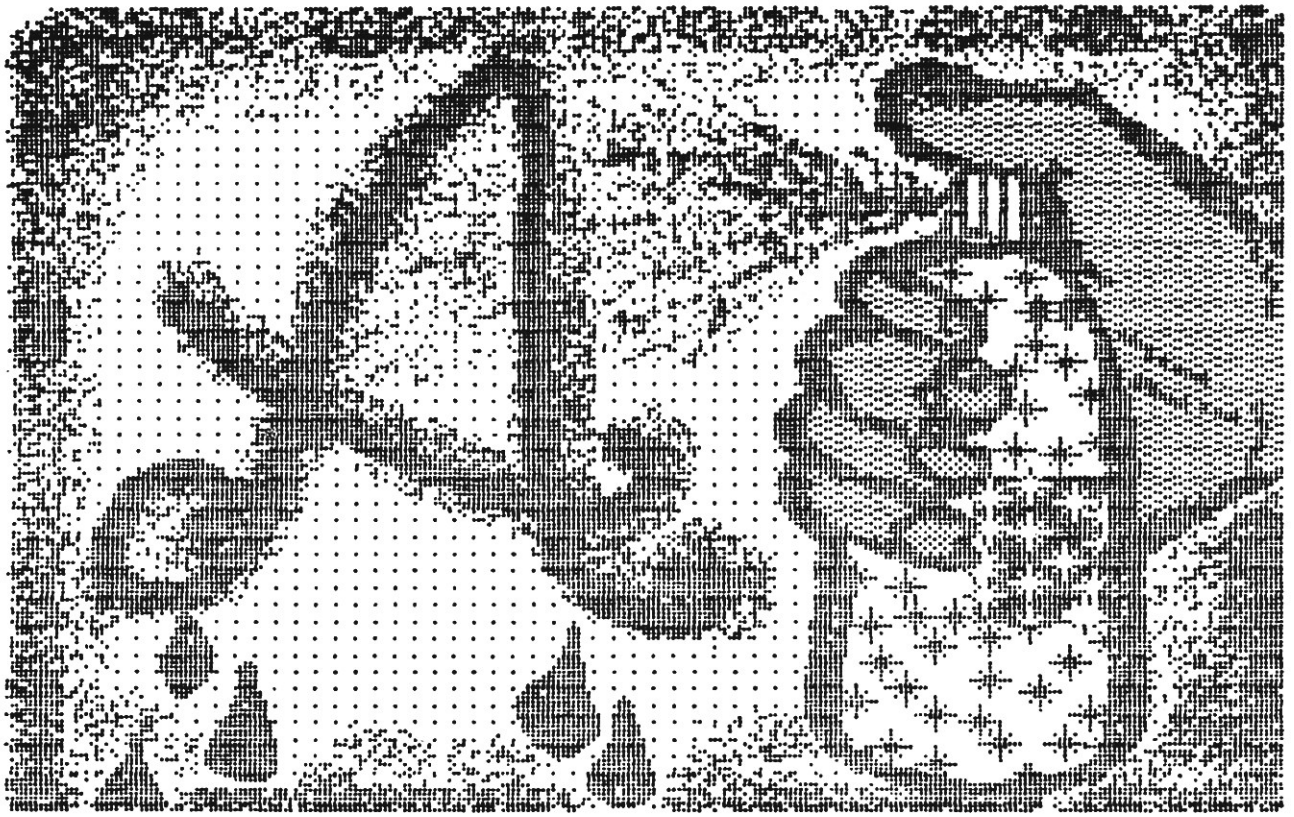
tintasugaras nyomtatók

Egy másik, szintén nem érintéses nyomtatási lehetőséget valósítanak meg a *tintasugaras nyomtatók.*

A tintasugaras nyomtatók mai sebessége nagyjából megegyezik a közepes teljesítményű sornyomtatók 900–1000 sor/perces sebességével, de sokkal szebben írnak, csendesek, kisebbek, grafikára kiválóan használhatók. Áruk már most összemérhető a közepes sebességű érintéses nyomtatók árával, s bizonyára idővel ez az ár is csökkenni fog.

Az egy vagy több fúvókán keresztül porlasztott tintacseppeket a nyomtatandó betű vagy ábra formájának megfelelően el kell téríteni a papírra érkezés előtt. Ez is a tv képernyőjére rajzoló elektronsugár eltérítési elvéhez hasonló módon történik. A tinta porlasztását pedig vibrációval biztosítják. Már ma is többféle tintasugaras nyomtató van forgalomban. Áruk ezer dollár alatt van, vannak ötszáz dollár körüliek is, amelyek 150 karakter/s sebességgel nyomtatnak.

Úgy tűnik, a színes nyomtatáshoz a tintasugaras nyomtatók jelenthetik az egyik lehetőséget. Amióta ugyanis létezik színes monitor a személyi számítógépeknél, kimondott kimondatlan vágya minden felhasználónak a színes papírkimenet.



Mint a fénytanból ismeretes, három alapszín, itt a sárga, a kék és a bíbor, elegendő bármilyen szín előállításához. Azonban a papír fehér volta miatt itt a fekete színre is szükség van. A feladat „mindössze" annyi, hogy a papír kívánt pontjára el kell juttatni a kívánt színt előállító alapszínek egy-egy cseppjét. Ez elvileg négyfúvókás tintasugaras nyomtatóval lehetséges. Persze a színes nyomtatás kisebb sebességű, mint a fekete, hiszen egy képpont több cseppből áll elő. A színes nyomtatók mintegy 10 karakter/s sebességgel képesek — jelenleg — működni. (A színes nyomtatáshoz a nyomtató berendezésen kívül megfelelő vezérlőprogramok is szükségesek.) A színes nyomtatók ára pillanatnyilag 1500—2000 dollár körül van. Ez az ár is várhatóan csökkenni fog az idő múlásával, és remélhetően lassan forintban is megadható lesz, ami azt jelenti, hogy a hazai piacon is megjelenik.

Színes nyomtatott képet elő lehet állítani olyan mátrixnyomtatóval is, amelyben több színű festékszalag biztosítja az alapszíneket. Ez is lassú, s ráadásul nem is olyan jó minőségű, mint a tintasugaras kimenet. Igaz, hogy olcsóbb.

És talán valahol már álmodozik az a szerelmes ifjú titán, aki elhatározta, hogy programozható illatú nyomtatót készít kimondottan szerelmesek számára. Talán még piaca is lesz a termékének.

22.

LEMEZTELENÍTVE NEM MEGY!
(Mágneslemezek)

A számítógépeket hosszú ideig főként tudományos, műszaki jellegű matematikai feladatok megoldására alkalmazták. Egy számítógéppel dolgozó tudósnak, mérnöknek általában jóval kevesebb gondot okozott a felhasználni kívánt adatok formája és tárolása, mert az alkalmazások többségében viszonylag kevés adattal dolgozott. A hangsúly a műveletek gyorsaságán és a hatékony matematikai algoritmusok kidolgozásán volt. A számítógépeknek nagy tömegű adatfeldolgozási feladatokra való felhasználása azonban szükségessé tette megfelelő nagyságú tömegtárolók és a rajtuk tárolt adatok kezelésére alkalmas szoftver kidolgozását.

központi tár

A *központi tárban* (operatív memóriában) levő adatokkal a processzor azonnal, közvetlenül tud aritmetikai vagy logikai műveletet végezni. A gép annál kényelmesebben használható, annál egyszerűbben programozható, minél több utasítás és adat helyezhető el egyidejűleg a központi tárban, azaz minél nagyobb annak a tárolókapacitása. Ennek a tárnak a méretét azonban műszaki, de főleg gazdasági okokból általában nem növelik egy bizonyos határ fölé. Ahhoz, hogy egy számítógéppel a ma használatos programtechnikai eszközök segítségével egyáltalán dolgozni lehessen, legalább 256 kilobájtos memóriára van szükség. A házi használatra szánt számítógépekben általában 16–64 kbájtos memória van, ezért ezek csak korlátozott programozási lehetőségekkel (pl. Basic) rendelkeznek. A nagy sorozatban gyártott, általános célú nagygépeknél jelenleg a memória 16 millió, mikrogépeknél 1–2 millió bájtig bővíthető. A műveleti sebesség meggyorsítása érdekében még ennek a központi tárolónak is van egy olyan úgynevezett gyorsregiszterekből felépített része, melyben a gyakran használt adatokat célszerű elhelyezni. Ha a műveleteknek akár csak az egyik operandusa is egy ilyen, drágább felépítésű, bonyolultabb tárolóelemben szerepel, akkor a művelet sokkal gyorsabban végezhető el, mint hogyha a központi tárolóból kellene a szükséges adatot vagy adatokat elővenni.

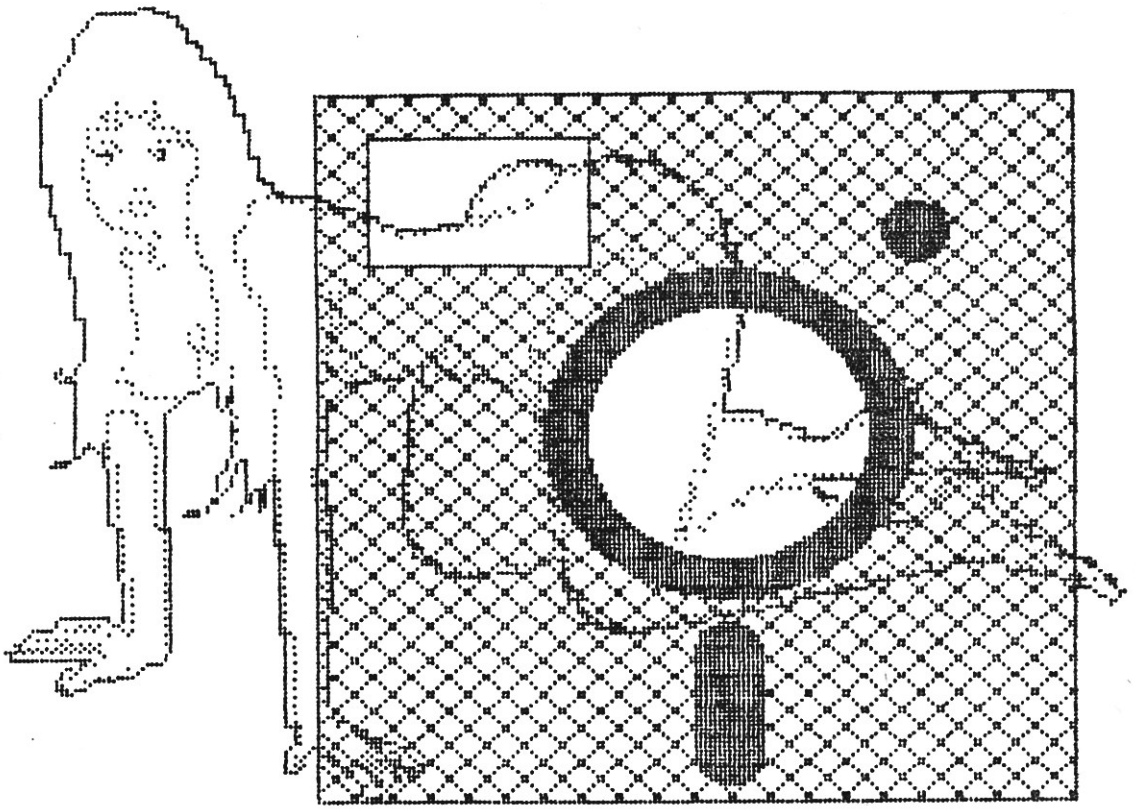
A számítógépeknek a vállalati, kereskedelmi életben, államigazgatásban való alkalmazásával azonban olyan nagy mértékben megnőtt a felhasználandó adatok mennyisége, hogy erre a célra külön tárolótípusokat kellett kifejleszteni, melyeket elfogadható áron alkalmassá lehet tenni a szükséges számú adat biztonságos tárolására és megfelelő gyorsaságú elérésére. (Például egy országos gépjármű-nyilvántartásban nem tudni előre, hogy éppen melyik gépkocsit kell azonosítani rendszáma, vagy esetleg színe, típusa, meg a baleset helyszínén talált egyéb nyomok alapján.) Nagy adatmennyiség hatékony tárolására leggyakrabban a különféle fajtájú mágneslemezeket használják. Gyakorlatban egy adott számítógép konfigurációnak a hozzácsatolt illetve hozzá csatolható lemezek és ezáltal a rajtuk tárolható adatoknak a mennyisége ugyanolyan alapvető jellemzője, mint a központi tároló nagysága vagy a processzor sebessége. Lemezek nélkül egy kis személyi számítógép is csak korlátozottan használható. Ilyen lemeztelenített géppel ma már nemcsak, hogy nem illik, de nem is lehet a piacon megjelenni.

A feldolgozáshoz nem állandóan szükséges adatok és programrészek közül azokat, amelyek nem férnek be a központi tárolóba általában mágneslemezen, időnként mágnes-

háttér-tárolók

szalagon tárolják és csak akkor olvassák be a memóriába, amikor éppen szükség van rájuk, olyan adatok helyére, amelyek a feldolgozás azon szakaszában már nem kellenek. Ezeket a tárolókat *tömeg-* vagy *háttértárolóknak* nevezzük.

Ezekre az operatív tártól függetleníthető tárolókra nemcsak azért van szükség, mert a nagy adatmennyiséggel dolgozó programok adatai nem férnek be egyidejűleg az operatív tárba, hanem azért is, mert ugyanazt a számítógépet, – ha időben nem is egyszerre, de több egymástól független feladat megoldására is használják. A sok programot és a hozzájuk tartozó adatokat már semmiképpen nem lehet elhelyezni a központi tárban. Nemcsak annak korlátozott mérete miatt, hanem azért sem, mert az operatív tár tartalma a gép kikapcsolásakor törlődik. Újraindításkor tehát valamilyen módon, de lehetőleg gyorsan, egyszerűen és hibamentesen be kell juttatni a gépbe a korábban megőrzésre eltett programokat és esetleg az azokkal előállított eredményeket is.



A beolvasás természetesen lényegesen megnöveli a műveletek elvégzési idejét, hiszen az adatokat előbb meg kell találni a tömegtárolón, majd be kell olvasni a központi tárba. (Ugyanígy lelassul a feldolgozás, ha a (rész)eredményeket ki kell írni a háttértárba.) A megtalálási idő függ attól, hogy milyen jól szerveztük az adatainkat és mennyire van szerencsénk. Ha az adatokat ugyanolyan sorrendben kívánjuk feldolgozni, mint ahogyan a tömegtárolón elhelyezkednek, akkor a megtalálási idő igen csekély, nagy gépeknél folyamatos feldolgozásnál ezredmásodperc nagyságrend körül mozog. Más esetekben azonban egy adat előkeresése mágneslemezeknél átlagosan néhány századtól néhány tizedmásodpercig terjed, míg mágnesszalagnál 2–3 percig is eltarthat. Ez az oka annak, hogy a mágneslemezek magasabb áruk ellenére is egyre inkább kiszorítják a mágnesszalagokat.

soros és közvetlen hozzáférésű tárak

Az elmondottak alapján következik, hogy a háttértárakat az adatok hozzáférhetősége szempontjából két nagy csoportba oszthatjuk, a *soros* és a *közvetlen hozzáférésű tárakra.* A soros hozzáférés azt jelenti, hogy a tárolt adatokhoz csak a fizikai elhelyezkedés sorrendjében lehet hozzáférni. Ezen tárolási módot leggyakrabban *mágnesszalagok, mágneskazetták* segítségével valósítják meg. A gyakorlatban a nagyszámítógépekhez több száz

méter hosszúságú, orsóra feltekercselt mágnesszalagokat használnak, melyeken jó helykihasználás esetén 20–40 millió karakter is tárolható. A személyi számítógépekhez a kisebb kapacitású, de jóval kisebb helyet elfoglaló, könnyebben kezelhető mágneskazettákat alkalmazzák. A tárolt adatmennyiségnek a kiírása illetve a beolvasása nagyszámítógépen 2–3 percet, személyi számítógépen 10–20 percet is igénybe vehet. Az utóbbi ilyenkor ezen idő alatt általában más feladatokat nem tud elvégezni, a nagy adatmennyiség átvitele hosszú ideig lefoglalja az egész berendezést. Nagyszámítógépeknél a mágnesszalagra írást ugyanúgy mint a más külső tárolóra történő adatátvitelt a bemeneti/kimeneti processzorok (lásd 9. és 12. fejezet!) önállóan bonyolítják, így a központi processzor más feladatok megoldásán tovább tud dolgozni. A mágnesszalagra- és mágneslemezre történő viszonylag hosszú ideig tartó írás nem zavarja a számítógép más irányú tevékenységét.

mágnesszalagok, mágneskártyák, mágneslemez

Egy számítógéphez általában több tömegtároló is hozzákapcsolható. Így az egyidejűleg elérhető adatok száma már közepes gépeknél is több százmillió bájtot tehet ki. Ráadásul a legtöbb tömegtárolón cserélhetők az adathordozók. Másfajta feldolgozáshoz más adathordozókat lehet a géphez csatlakoztatni. Ezáltal a gép tárolókapacitása – ha nem is egyidejű a felhasználási lehetőséggel – de szinte korlátlanra növelhető.

A személyi számítógépeknél általánosan használt *mágneslemez,* az úgynevezett hajlékony lemez hasonlít egy hanglemezhez. Erről a következő fejezetben bővebben is beszélünk. Nagyobb gépeknél egy lemezcsomag több egymás fölé helyezett lemezből áll, melyeknek 10–20 mágnesezhető felülete van. A felületen 100–200 koncentrikus mágnesezhető kör alakítható ki. Ezeket a köröket sávnak vagy pályának nevezzük. Az adatok ezeken a sávokon helyezkednek el. A mágnesezettség egyik iránya a 0, másik iránya az 1 számjegyet jelenti. A leolvasás illetve a felírás a mágnesezettség leolvasásával illetve felülírásával történik oly módon, hogy az író-olvasó fejet a megfelelő sáv fölé visszük, úgy, mint amikor lejátszáskor a hanglemezen a tűt a megfelelő zeneszám fölé rakjuk. Ez a mozgatási idő 0,01–0,2 s nagyságrend körül van. A lemez percenként több ezres fordulatszámmal forog.

Így legkésőbb egy fordulaton, néhány századmásodpercen belül a fej alá kerül a kívánt adat. Ekkor, ahogy a forgás következtében elhalad a fej alatt, a mágneses tartományok bitenként kiolvashatók onnan, – illetve beírhatók oda.

A lemezcsomagokon tárolható bájtok száma – típustól függően – néhány milliótól több száz millióig terjed.

23.

FORMÁS LEMEZEK – LEMEZFORMÁK
(A személyi számítógép háttértárai)

A személyi számítógépek egyre nélkülözhetetlenebb tartozéka a *hajlékonylemez-meghajtó.* Ennek a mondatnak a jelentőségét csak az tudja igazán értékelni, aki már sokat unatkozott egy magnó mellett, miközben a kazettáról szép komótosan töltődött egy-egy program az otthoni számítógépbe.

Csak sajnos a lemezmeghajtó nem olcsó mulatság. Az otthoni számítógépekhez kapható különálló meghajtók ára nagyjából az alapgép árával azonos, de inkább 1,5–2-szerese. A professzionális gépeknél a lemezmeghajtó (egy vagy kettő) a géppel közös házikóban van elhelyezve, így ezeknek látszólag nincs külön ára, de valójában – az egységhez hasonlóan – be van építve az alapgép árába.

hajlékony-lemez

Mindez persze nem gátolhat bennünket abban, hogy közelebbről is szemügyre vegyük a *hajlékonylemezt.* Ezt mindenki egyszerűen csak flopinak nevezi, az angol „floppy diskette" = hajlékony lemezecske „becézéseként", talán azért mert a magyar elnevezés túl hosszú.

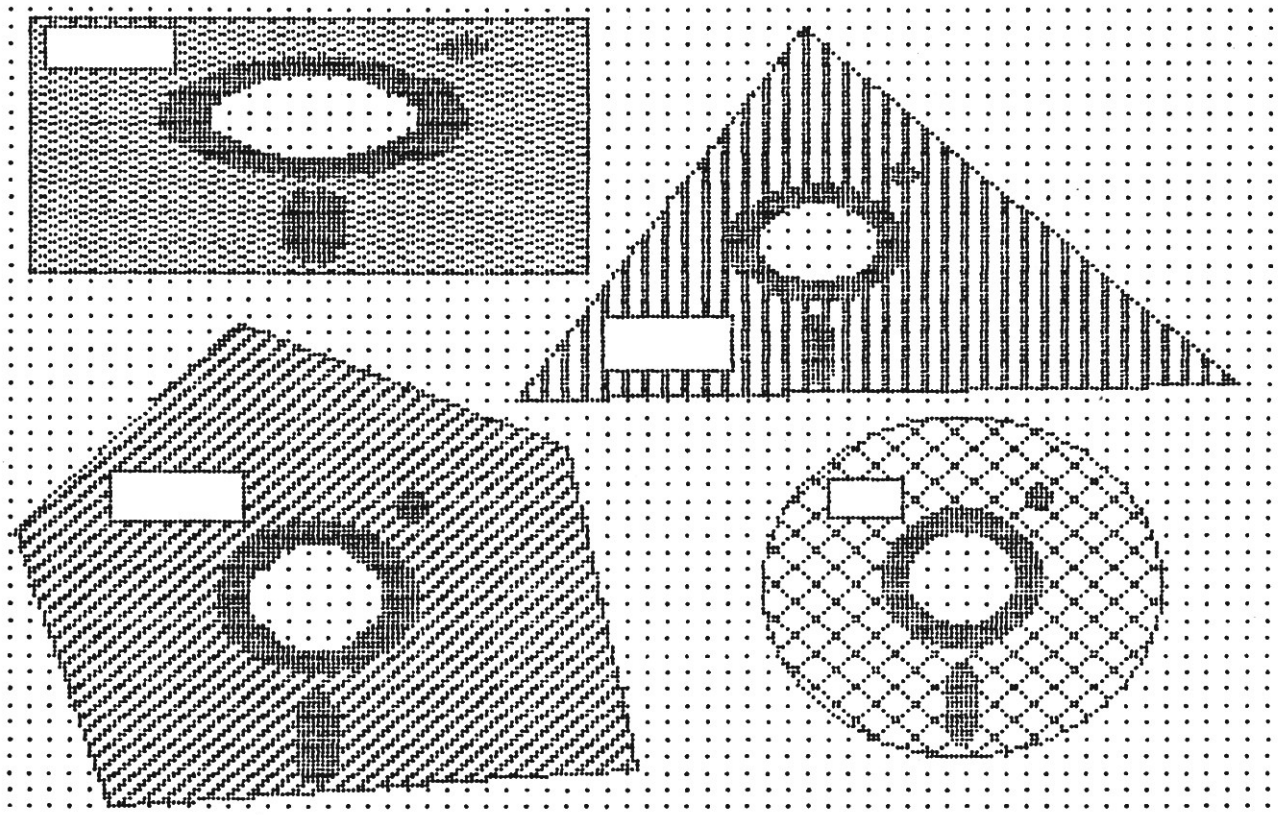
A hagyományos számítógépek mágneslemez csomagja volt a minta, amiből a mikroszámítógépek közvetlen elérésű külső tárolója megszületett.

Kezdetben a lemez átmérője 8 hüvelyk (kb. 20 cm) volt. Ezt hamar felváltotta a könnyebben kezelhető, 5 1/4 hüvelyk (133,4 mm) átmérőjű hajlékony lemez. A lemezek egy megfelelően kialakított védőtokban vannak elhelyezve. Külön papírtasak szolgálja a védelmet a használaton kívüli időszakokban. (Megjelentek a merev tasakolású 3,5 hüvelykes mikrolemezek is).

A meghajtóba megfelelően elhelyezett lemez a használat során a műanyag tokban forog. A forgás sebessége általában 360 fordulat percenként. A lemez, amely felületén mágnesezhető anyaggal van bevonva néhány helyen „kilátszik" a tokból. Legfontosabb az a nyílás, amelyen keresztül a meghajtóba helyezett lemezfelülethez az író-olvasó fej hozzá tud férni. Ezt az ovális lyukat fejnyílásnak nevezzük. Egy másik, kör alakú nyílás a lemez helyes elhelyezésének ellenőrzésére szolgál. Lehetőség van arra, hogy a lemez tartalmát fizikailag megvédjük az esetleges felülírástól. A lemeztok oldalán egy bevágás szerepel, ez teszi lehetővé a lemezre való írást. Ha ezt a bevágást leragasztjuk, a lemez csak olvasásra használható. Igen erősen javasoljuk, hogy azokat a lemezeket, amelyeknek tartalmát hosszú ideig változatlanul meg kívánjuk őrizni, tegyük ily módon „írhatatlanná". Ha egyszer mégis kívánnánk rájuk írni, akkor egyszerűen eltávolíthatjuk ezt a leragasztást.

Az információk elhelyezése a lemezen sávok mentén történik. A lemezen található sávok számát befolyásolja, hogy milyen típusú meghajtón kerülnek használatba. A 40 sávos hajlékonylemezt kezelő meghajtók igen elterjedtek, ezért az ezeknél használatos lemezeket 40 sáv méretűre kell formázni (lásd az 50. fejezetet!). Találkozhatunk ennél több (pl. 80) sávot kezelő meghajtókkal is. Ugyanígy nem egységes az írási sűrűség sem. Mindez természetesen végső fokon a lemez kapacitását befolyásolja, valamint azzal a kellemetlenséggel is jár, hogy különböző típusú gépek nem mindig tudják egymás lemezeit elolvasni (pl. a Commodore 64 és az IBM PC).

Jelentősen sikerült a lemezkapacitást növelni, amikor az egyoldalas lemezmeghajtók mellett megjelentek azok a meghajtók, amelyek egyidejűleg képesek a két lemezoldalt kezelni, amihez persze olyan lemez is kell, melynek mindkét oldala használható. Az információ helyének azonosítása ebben az esetben az oldal, a sáv és a sávon belüli szektor megadásával lehetséges. Többféle kapacitású lemez van forgalomban. Az egyoldalasoknál 160–180 kbájtos, a kétoldalasoknál 320 kbájttól 1.2 Mbájtig terjednek. A hajlékony lemezen is létrehoz és adminisztrál a működtető rendszer egy tartalomjegyzéket, amelynek tartalma (a lemezen lévő programok vagy adatállományok neve) bármikor megjeleníthető a képernyőn. Mivel egy-egy felhasználó sok lemezzel szokott dolgozni, ezért fontos, hogy a lemezeken kívül is legyenek tájékoztató információk. Ezt felragasztható címkékkel lehet biztosítani, de ezen kívül célszerű a számítógéppel is „megcímkézni" a lemez elejét (lásd az 50. fejezetet!).



A hajlékonylemez borítóján hasznos tudnivalókat találunk arra vonatkozóan, hogy mit ne csináljunk egy lemezzel. Így például ne fogjuk meg ott, ahol kilátszik a tasakból a lemez felülete, ne tegyük mágnes közelébe, ne írjunk a felragasztott címkére kemény íróeszközzel (pl. golyóstollal), s bármennyire is hajlékony a lemez, ezt a tulajdonságát ne próbálgassuk ellenőrizni.

A meghajtók átviteli sebessége jobb minőség esetén eléri a 250 kbit/s-ot. A lemezen történő keresési idő néhány tizedmásodperc nagyságrendű.

A professzionális gépek megjelenésével a hajlékony lemezen biztosított háttérkapacitás kevésnek bizonyult. Ezt elvileg a meghajtók számának növelésével ellensúlyozni lehetett volna, de a hajlékonylemez-meghajtók is a drága perifériák közé tartoznak.

Kifejlesztettek zárt felépítésű (a lemezek és az olvasófejek egy műszaki egységet alkotnak) háttértárakat is. Ezeket nevezzük *winchestertáraknak*. Vannak olyan winchestertárak, amelyek bele vannak építve az alapgépbe. Ezek kapacitása általában 10–20 Mbájt, külső méretei megegyeznek a hajlékonylemez-meghajtó méreteivel. Az azonos helyen

winchester-tárak

elhelyezhető nagyobb (több mint tízszeres) kapacitást a winchester szerkezeti felépítése teszi lehetővé. Ezek a tárak légmentesen zárt tokban vannak, az olvasó/író fejek távolsága a lemezfelülettől emiatt sokkal kisebb (nem kell a lemezfelületet ki-be mozgatni, nem keletkezik szennyeződés a felületen). Mindez sokkal nagyobb írássűrűséget tesz lehetővé. Egyes winchestertárak (és meghajtóik) külön egységként kerülnek forgalomba, ezek kapacitása már 80–100 Mbájt körüli. Várhatók még ennél nagyobb kapacitású winchestertárak is. E tárak megjelenésével a professzionális személyi számítógépeknek nemcsak a háttérkapacitása nőtt meg, hanem egyes esetekben a feladatmegoldó sebességük is, mivel a winchesteren lévő adatok elérési ideje sokkal kisebb, mint a hajlékony lemezen lévőké.

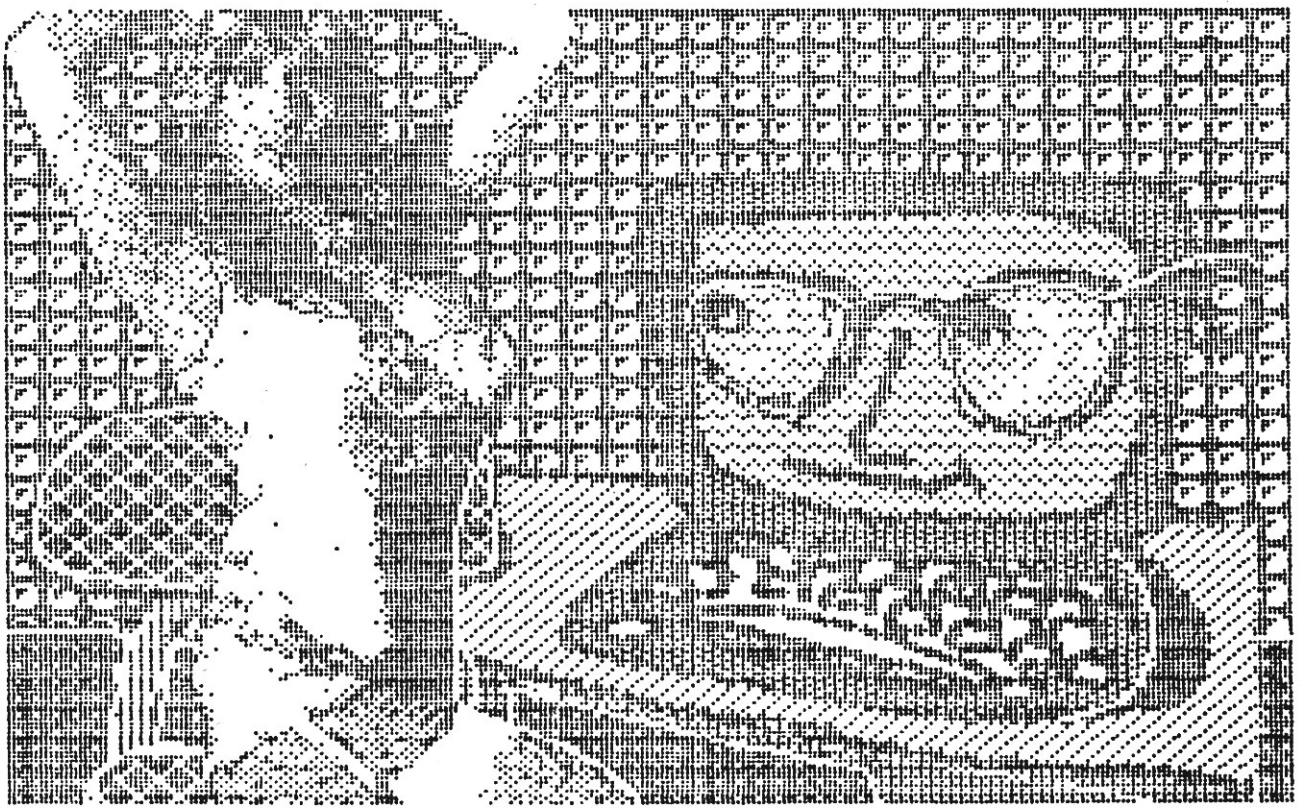
Akár hajlékony lemezt, akár winchestert kívánunk használatba venni, a lemezfelületeket elő kell készíteni a tárolni kívánt információk fogadására. Ezt a munkát természetesen program végzi el, de ez már egy másik történet. Azokról a váratlan „élményekről", amelyek egy lemez formázásakor érhetnek bennünket az 50. fejezetben írunk majd.

24.

MIT LÁTNAK SZEMEIM!
(Optikai jelolvasók)

Mit látnak szemeim — kiáltott a kezdő író első regényében a vak koldus. Ez komikus, mert a látás fogalmát a szemhez kapcsoljuk. (Ott a szerkesztő át is íratta szerzőnkkel ezt a mondatot.) De miért beszélünk látásról a számítógépek esetében?

Amióta a számítástechnika bevonult az adatfeldolgozás területére (és ez már az ötvenes években megtörtént), máig meglévő probléma a feldolgozandó adatoknak a gépbe juttatása. Ennek egyik, ma is létező formája az adatok előzetes, külön munkafolyamatot jelentő rögzítése. A rögzítés régebben papír alapú adathordozókra, lyukszalagokra, majd később lyukkártyákra, manapság valamilyen mágneses adathordozóra (mágnesszalag, hajlékony lemez) történik. Ez a tevékenység azonban időigényessége mellett nagyon sok hibalehetőséget is hordoz, hiszen az ember mindig valamilyen hibaszázalékkal tevékenykedik.



Az igazán jó megoldás az, ha a számítógép közvetlenül azt az alapbizonylatot tudja elolvasni, amelyre az adatokat megszületésükkor rögzítették. (Ilyen alapbizonylatok például egy raktárban a kivételezési jegyek, vagy a szállítólevelek.) És máris a látásnál vagyunk, hiszen a gép akkor tud ilyen papírokat olvasni, ha „látja" azt, ami rá van írva.

Toshiba levél-osztályozó

Kevesen gondolnak arra, hogy a Magyar Postánál működő *Toshiba levélosztályozó* berendezésben is számítógép lapul, s éppen az irányítószámok felismerését végzik a mikroprocesszorok. Ez a felismerés a számjegyek képének digitalizálásával lehetséges. A digitalizált kép már tárolható számítógépben, s ez azt jelenti, hogy bármilyen szempontból feldolgozható, osztályozható. A levélosztályozó több speciális célszámítógép együttes munkájával oldja meg feladatát. Mindennapos tapasztalatunk, hogy az irányítószámok formájára vonatkozóan a Posta nem tesz kikötéseket, mindössze azt kéri, hogy minden kézzel írott számjegy a neki megfelelő piros téglalapban legyen.

optikai bizonylat-olvasók, OCR jelek

Az Európai Számítógépgyártók Szövetsége már 1961-ben bizottságot hozott létre az *optikai bizonylatolvasók* jelkészlet-szabványának kialakítására. A többféle javaslat közül az OCR–A és az OCR–B elnevezésű változatok váltak elterjedtté. (OCR = Optical Character Recognition = optikai úton történő jelfelismerés).

A szabvány jelkészlet betűit írógépek írókarjaira vagy gömbfejeire szerelték, s ezekkel az írógépekkel állították elő a gép által is olvasható bizonylatokat. Ha a számítógép sornyomtatója is ilyen betűtípussal volt ellátva, mód nyílott a nyomtatott szöveg visszaolvastatására.

Az *OCR jelek* olvasását végző berendezés egyik legfontosabb része a képfelbontó volt, amelyben egy-egy karakter letapogatása fotodiódák segítségével történt. A másik fontos része e berendezésnek a felismerő egység, amely a már érzékelt jelet összevetette a szótárában lévő lehetséges jelalakokkal, s kiválasztotta azt, amellyel az olvasott megegyezett. (Külön problémát jelentett, ha nem volt olyan jel a szótárban, amellyel a beolvasott tökéletesen egyezett. Ez adódhatott pl. abból, hogy az írógépen egy-egy betű megkopott. Erre is volt megoldás: előre meghatározott szintű hasonlóságot a rendszer elfogadott.) E két egységet egészítette ki a bizonylattovábbító és a kimeneti egység. Ez utóbbi a felismert jelet pl. mágnesszalagon rögzítette, az előbbi a bizonylatok fizikai mozgatását biztosította.

Miért írjuk mindezt múlt időben? Mert a technikai fejlődés más irányoknak adott nagyobb teret. A már említett digitalizálás vagyis képek felbontása diszkrét, számítógépben kódolható és így eltárolható jelek, angolul digitek sorozatára, (a digit szó a töve a digitalizálni igének is) ma már lehetővé teszi, hogy pl. egy teljes szövegoldalt mint egyetlen „képet" a számítógép minden nehézség nélkül elolvasson, megjelenítse a képernyőn, és biztosítsa a felhasználót, hogy azon bármit változtasson. Mindez már messze túlmutat az adatbevitel problémakörén, s új lehetőséget nyitott a számítástechnika számára pl. a térképészetben, építészeti tervezésben, s más grafikus információkat használó alkalmazásokban.

S mivel a digitalizáló berendezések ára ma már nem magas, ezek az alkalmazások egyre inkább terjedhetnek.

A látó számítógép mellett nemsokára megjelenik a beszélő számítógép is. Gondoljuk csak el, hogy vak emberek számára mekkora segítséget jelent egy ilyen olvasó-beszélő berendezés.

Az optikai adatérzékelés egy nagy jövő előtt álló területe a vonalkód és alkalmazása. Erről a következő fejezetben szólunk.

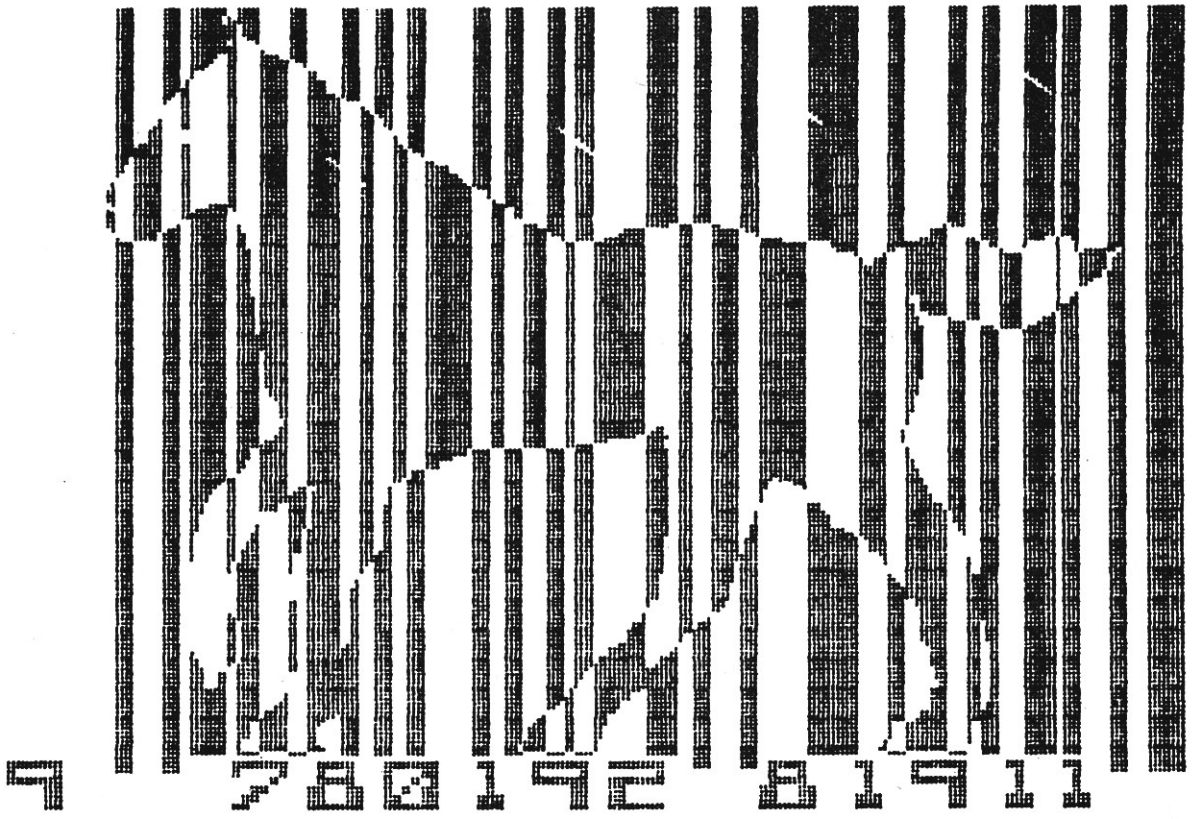
25.

VONALASAK LESZÜNK ! (?)
(Vonalkódok)

Melyik háziasszony nem bosszankodott még egy-egy bevásárlás után, amikor észrevette, hogy megint néhány forinttal többet számoltak a pénztárnál. És melyik pénztáros nem bosszankodott még, amikor a vevő kosarában lévő italos üvegen, vagy húskonzerven nem találta az árat, és hiába emlékszik rá, mert éppen tegnap változott meg.

Gyorsan változó világban élünk, s e gyors változások a fogyasztói árakon is tapasztalhatók. Az árukészlet új árcímkékkel való ellátása – mégha bármilyen korszerű árazó gépek is állnak rendelkezésre – sok munkát igénylő, időtrabló feladat. Ezért a bosszankodók közül ne felejtsük ki a bolti eladókat, akik a gyakori címkézést, meg az ilyenkor elengedhetetlen leltározást végzik.

Az üzletvezető pedig visszasírja azokat a régi szép időket, amikor minden árucikk hatóságilag rögzített – s ezért hosszú ideig változatlan – áron került forgalomba. De azért mégsem gondolja senki, hogy ez az üzletvezető vonalas.



Vonalkód

„Vonalasak" viszont azok a termékek (csokoládék, konzervsörök, magnó- és videókazetták stb.), amelyeket külföldön csomagoltak. És vannak már olyan magyar csomagolású árucikkek, amelyeken szintén felfedezhetők a *vonalkódok.*

Magyarország 1984. január 1-je óta tagja az EAN-nek, az Európai Vonalkód Rendszernek. Négy-öt egymástól különböző vonalkódrendszer terjedt el nemzetközi szab-

ványként. A már említett fogyasztási cikkekre nyomott szabvány az ún. EAN/UPC rendszer. Említést érdemel még a polgári repülésben a csomagok illetve a jegyek azonosítására kidolgozott IATA kódrendszer.

EAN/UPC kód

Az *EAN/UPC kód* felépítése alapján azonosítható a terméket kibocsátó ország (hazánk az 599 kódra „hallgat" illetve ezen a vonalkódon látható), a termelő vagy importáló vállalat, valamint maga a termék. Az utolsó kódjegy pedig egy ellenőrző szám, amelynek segítségével ellenőrizhető, hogy a leolvasás sikeres volt-e.

Miért nem számjegykódok formájában jelennek meg ezek az információk az árun? Azért, mert a vonalkód olvasását a gép kisebb hibával képes elvégezni, mint a számjegykódokét.

Ha a vonalkód rendszer előnyeit meg akarjuk érteni, menjünk el (legalább is gondolatban) egy olyan üzletbe, ahol használják ezeket a kódokat. (Bár nem nagy számban, de van már ilyen üzlet hazánkban is.) A vevő a polcokról összeszedi mindazt, amire szüksége van. A pénztárnál a pénztáros minden árucikket elhúz egy fényforrás előtt, úgy, hogy a fény a vonalkódot éri. (Esetleg egy leolvasó „ceruzát" húz végig a vonalkód előtt.) A pénztárgépben elhelyezett mikroprocesszor a kód alapján azonosítja az árucikket, memóriájában megkeresi az adott cikk árát (esetleg a megnevezését is), s ezt követően a blokkra kinyomtatja a fizetendő részösszeget. Az utolsó tétel után pedig a számla végösszegét. A pénztárosnak csak a pénz átvételével kell foglalkoznia. Bármilyen árváltozás esetén csak a számítógépbe rögzített árat kell (egyetlen helyen) változtatni, s nem kell minden egyes árucikkre felcímkézni.

S ha mindez még nem ellensúlyozná azokat a költségeket, amelyek a vonalkódnak az árucikkre való felvitelét jelentik, itt vannak a további előnyök:

- a pénztárgépben lévő mikroszámítógép összeköttetésben állhat egy központi géppel, amely pontosan regisztrálja, hogy melyik árucikkből mennyi fogyott,
- ebből következik, hogy a napi forgalmi kimutatás nagyon gyorsan előállítható árucikkmélységben,
- amiből kiderül, hogy miből mekkora mennyiséget kell az eladótérbe juttatni az éjszaka folyamán. (Ahol ilyen üzletek vannak, ott már sikerült megvalósítani a vevőket nem zavaró éjszakai szállítást),
- probléma nélkül elkészíthető a rendelési lista, amelyet a szállítóknak kell elküldeni.

A vonalkódon alapuló számítógépes rendszer tehát szinte minden adminisztrációt levesz az üzletvezető válláról. Ez teszi népszerűvé. Amitől mégsem terjed olyan mértékben mint kellene, az a beszerzési költség. Egy ‚tíz pénztáras' rendszer ára, a központi professzionális számítógéppel együtt meghaladja az ötmillió forintot. És mindez csak akkor működik megfelelően, ha *minden egyes* árucikken rajta van a saját vonalkódja.

Ehhez nagyon fejlett csomagolási kultúra szükséges. De addig sincs minden veszve. A vonalkódot ugyanis egy ügyes szerkezet segítségével a kereskedő is ráírhatja áruira. Kaphatók ugyanis speciális, vonalkódot nyomtató berendezések, amelyek pc-hez kapcsolhatók, s a számjegyekből álló kódból előállítják a megfelelő vonalkódot egy öntapadós címkére.

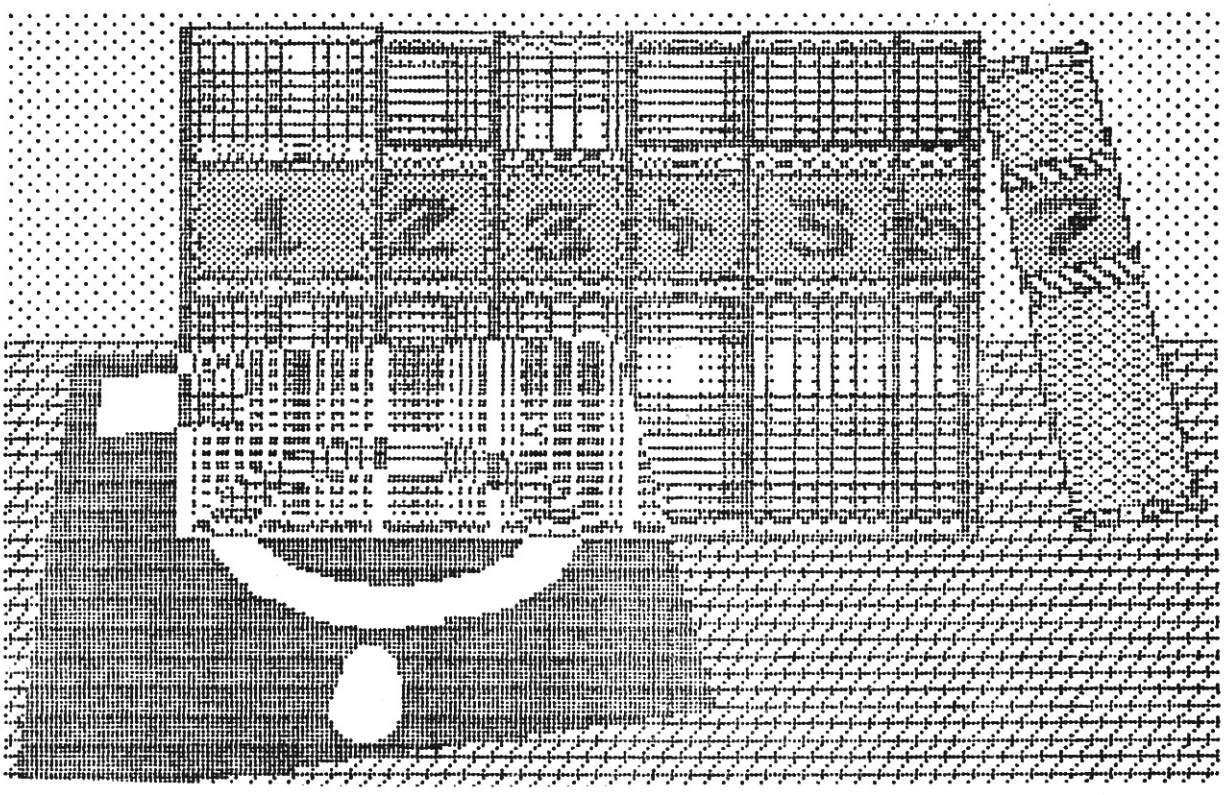
De ez mégsem az igazi megoldás, hiszen költségesebb. Ha a csomagolási kultúránk felnő a feladathoz, s a jelenleginél többen belátják, hogy a vonalkódba befektetett pénz idővel megtérül, büszkén hirdethetjük országnak világnak, hogy mi is vonalasak vagyunk.

26.

LEXIKON EGY OLDALON
(Optikai tárak)

Ha ezen könyv szerzői és kiadója igazán korszerűen dolgoznának, akkor e terjedelmes kötet helyett az olvasó egy ezüstösen fénylő, a digitális hanglemezhez hasonló kicsiny korongot tartana a kezében, melyet a személyi számítógépéhez illesztett CD (compack disk) olvasóba betéve, azonnal lehívhatná a képernyőjére vagy kinyomtathatná a nyomtatójára a kívánt fejezeteket, vagy a keresett témakörrel foglalkozó részeket.

Sajnos sem a magyar könyvkiadás, sem a magyar olvasóközönség technikai ellátottsága még nem tart itt. Amíg nálunk évtizedek óta nincs megfelelő lexikon, addig a külföldi piacon már megjelent az első igazi számítógépes nagy lexikon. A húsz kötetes Grolier Academic American Encyclopedia egyetlen kompakt lemezen elfért, és ára mindössze 200 dollár, jóval alacsonyabb mint a könyv formájában 10 ezer lapot kitevő sorozaté. A valamivel több mint 100 megabájtnyi adatot tartalmazó lemeznek mintegy a felét a tényleges lexikális adatok foglalták el, míg a többin a visszakeresést szolgáló további segédadatokat helyezték el.



A hangtechnikába már néhány éve bevonult digitális lemezek, a *kompakt diszkek* (CD) elindultak, hogy meghódítsák a személyi számítógépek alkalmazóit is. Első bemutatkozásaik kiváló eredménnyel zárultak.

kompakt diszkek

A gyakorlatban is sikerrel alkalmazzák már az optikai tárak közül az egyszer beírt adatoknak akárhányszori újraolvasására alkalmas CD–ROM-okat. (A ROM = Read Only Memory = csak olvasható tár. A számítógép központi tárában játszott szerepéről már szóltunk a 13. fejezetben.)

optikai adattárolás

Az *optikai adattárolás* alapja a lézer. Az igen kicsi, szinte mikroszkopikus nagyságú területre fókuszálható, nagy energiájú koherens fénysugár hozza létre a lemez anyagában azokat a maradandó elváltozásokat (írja be az egyes biteket), melyek helyzetét olvasáskor a visszavert fénysugár erősségének változásából meg lehet állapítani.

Az adattároló a szokásos lemezekkel azonos méretű (kb. 12 cm átmérőjű), polikarbonát alapanyagú lemez. Erre egy vékony fényvisszaverő alumíniumréteget, vagy arany-, vagy platinaötvözetet és egy ezt védő fényáteresztő felületet visznek fel.

Beíráskor a nagy energiájú lézersugárzás hatására a lemez helyileg fölmelegszik, a fényvisszaverő réteg felpúposodik. Ezért itt a fény másképpen, jobban szétszórva verődik vissza, mint a sugárzás által nem érintett, sima felületről. Olvasáskor az olvasófej a visszavert fényben emiatt bekövetkező intenzitásváltozást észleli.

A sima és egyenetlen felületek váltakozásából – ugyanúgy, mint egy hanglemezen – egyetlen sáv alakul ki. Ennek hossza több kilométer, a menetek távolsága 1–2 ezred milliméter (mikromilliméter), míg a sima és egyenetlen felületek vastagsága és szélessége néhány tízezred milliméter körül van. Az adatok szervezése a hajlékony lemezéhez (lásd 23. fejezet) hasonló. Az adatsáv rögzített hosszúságú szektorokra van osztva. Minden szektor tartalmazza a saját pontos címét (leolvasáskor ennek azonosítása alapján ellenőrzik, hogy az olvasófej valóban jó helyen állt-e), és meghatározott számú (általában 2048) bájtnyi adatot. Az adatokat bitenként, szorosan egymás után írják fel. Az egyik lehetséges felírási mód a következő: Minden egyes bitnek a szektoron belül egy meghatározott hely felel meg, amelyet a bitnek a jelsorozatban elfoglalt helye szab meg. Ahol a bitsorozatban egyes van, olda ráirányítják a beíró, erős lézersugarat. Ennek hatására ott, az adott bitnek megfelelő helyen hődeformáció jön létre. Ahol viszont zérus áll, ott sima marad a felület. A végleges deformáció miatt ez a lemez csak egyszer írható, de elvileg akárhányszor olvasható, mivel az olvasó sugárzás intenzitása nem elég ahhoz, hogy a felületen változást okozzon. Egyetlen egy szabványos lemezen 550 Mbájt tárolható, ezért ez az eljárás igen eredményesen használható egyszer s mindenkorra változatlanul archiválandó adatok tárolására, könyvek szövegének kis helyen való elhelyezésére.

Hogy ennek az adatmennyiségnek a nagyságát érzékelhessük, egy szabványos gépelt oldal sortávolságtól és margótól függően 1300–2000 bájtot jelent, míg a szokásos méretű könyvek oldalanként 3–4000 bájtot tartalmaznak. Egy 12 cm átmérőjű lemezre tehát mintegy 600 db, egyenként 250 oldalas könyv teljes szövege elhelyezhető. A korszerű professzionális személyi számítógépekhez sokféle CD játszó hozzákapcsolható, és az eljárás széles körű elterjedésével az adatokat tartalmazó lemezek is egyre olcsóbbá válnak.

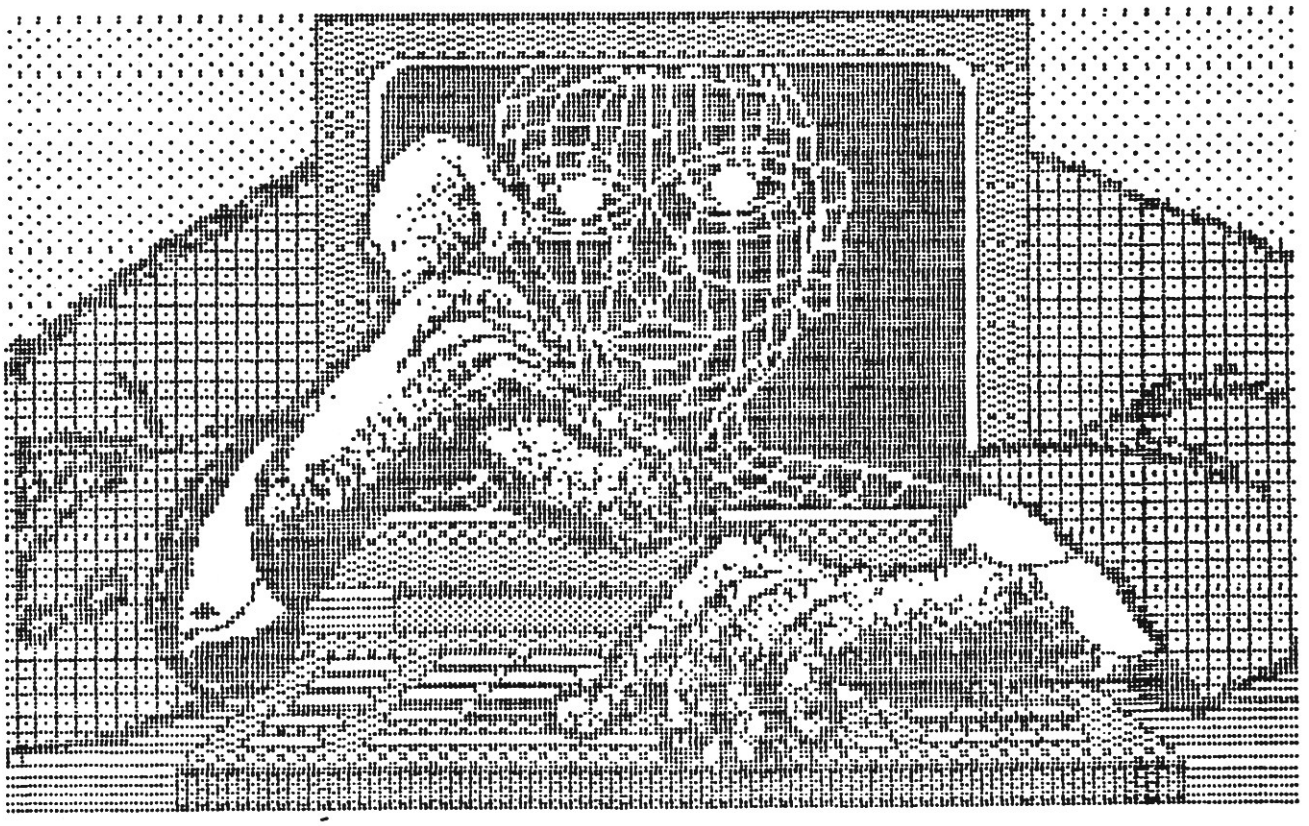
Iskolás gyerekeink – főleg az alsó tagozatosok – időnként hétrét görnyedve cipelik több kiló (ahogyan ők tanulják, helyesen, sok Newton) súlyú tankönyveiket. A jövő iskolájában az egész általános és gimnáziumi tananyag a kötelező olvasmányokkal együtt rajta lesz egyetlen kis lemezkén. Nem azért, mert addigra olyan kevés lesz a tanulnivaló, hanem azért, mert ilyen tömörített módon tudjuk majd tárolni az információt.

27.

BE VAGYUNK HÁLÓZVA
(Távfeldolgozó hálózatok)

Kezdetben volt a számítógép, amelyhez minden felhasználó hozzáférhetett. Hihetetlennek hangzik, de így van. Tudniillik nagyon kevés felhasználó volt, általában csak a mérnökök, matematikusok, fizikusok, akik egyben a gépek üzemeltetői és mindenesei voltak. Jóformán ott éltek a gépteremben. De amint a felhasználók száma elkezdett növekedni, az üzemeltetésért felelős főnök azonnal kiparancsolt mindenkit a gépteremből, diszpécsernek nevezett női cerberusokat (cerberákat) állítva a gép és a felhasználó közé. És elkezdődött a keserves korszak, amit closed shop-nak (zárt üzemnek) vagy kötegelt feldolgozási rendszernek neveztek. Ilyenkor a kisebb programozási hibák javítása is egy napig tartott. (Aki csokoládét vitt a cerberáknak, annak esetleg fél napig.) A javított programot a diszpécsereknél kellett leadni (lyukkártyán rögzítve), akik azt időnként beküldték a gépterembe. A futás eredményét szintén a diszpécserek tették egy e célra szolgáló polcrendszer meghatározott helyére.

Aztán megszületett a nagy ötlet — aminek megvalósítását a szüntelenül fejlődő technika tette lehetővé —, közvetlenül hozzáférni a géphez, anélkül, hogy belépnénk a gépterembe. Hogyan lehetséges ez?



A számítógépektől nem túl távol olyan készülékeket helyeztek el, amelyek adatbevivő billentyűzetből és képernyős, vagy írógépszerű megjelenítőből álltak. Ezek a készülékek a terminálok, amelyek vezetékeken (kábeleken) keresztül voltak összekötve a számítógéppel. A számítógépbe a kábelen át beérkező jeleket megfelelő hardver és működtető

időosztásos üzemmód

(operációs) rendszer fogadta a gép oldaláról. Végeredményben a gép és az ember működési sebessége közötti óriási különbség tette lehetővé, hogy egyidejűleg több felhasználó használhassa *időosztásos üzemmódban* ugyanazt a számítógépet. A terminálokon keresztül a programfejlesztési folyamat sokkal kényelmesebbé vált, mint a „beadom–visszakapom" szisztémával.

Az időosztásos rendszerben minden felhasználó saját könyvtárakkal rendelkezik. Ezek a könyvtárak mágneslemezen levő állományok, amelybe a különböző programok (ritkábban adatállományok) kerülnek. A programokhoz való gyors hozzáférést a könyvtári szerkezet biztosítja. A könyvtárban őrzött programok mindaddig megmaradnak, amíg maga a felhasználó ki nem törli. A bejelentkezni szándékozó felhasználót a rendszer ellenőrzi, mielőtt megengedné a bekapcsolást. Ez az ellenőrzés általában kétszintű. Egy lehetséges felhasználói azonosítót kell közölni a számítógéppel, majd ezt követően az azonosítóhoz tartozó jelszót. A jelszó titkos, begépeléskor nem is jelenik meg a képernyőn (vagy régebben nem volt elolvasható a terminál írógéppapírján). Minél fontosabb adatokhoz lehet hozzáférni a rendszerbe való belépés után, annál nagyobb hangsúlyt fektetnek a számítógép üzemeltetői a jelszók titkosságának megőrzésére. Az a felhasználó, aki érvényes azonosítóval és jelszóval rendelkezik, beléphet a rendszerbe és ezzel jogot nyer arra, hogy annak erőforrásait használja. Így módja nyílik arra, hogy létrehozzon programot valamilyen programnyelv felhasználásával, azt a géppel lefordíttassa, majd (sikeres fordítás és összeszerkesztés után) futtassa. Sikertelen fordítás vagy futtatás után azonnal elkezdheti a hibakeresést és hibajavítást. Ily módon egy-két óra alatt elvégezhető az, ami korábban napokig tartott.

Minél nagyobb a központi számítógép (ez itt most a központi tár méretet és háttértár kapacitást jelenti), s minél jobb működtető rendszerrel rendelkezik, annál több terminált tud kiszolgálni, anélkül, hogy a gép válaszideje számottevően nőne. Nagy egyetemeken pl. több száz terminálos rendszerek működnek.

Ez esetben a terminálok között lesznek olyanok, amelyek olyan távolra kerülnek a központi számítógéptől, hogy az alkalmazott kábeles összeköttetés már nem tudja biztosítani a jelek megfelelő átvitelét.

A nagyobb távolságra szánt jeleket ezért telefon- vagy telexvonalakon továbbítják. Ezek ugyanis általában már léteznek, de ha külön e célra kell is telepíteni, olcsóbb megoldást jelentenek. A telefon- illetve telexvonalakon történő kommunikációnak azonban van egy problémája. A számítógépes jeleket át kell transzformálni abba a frekvenciatartományba, amelyet ezek a vonalak megkívánnak. Ezt az átalakítást az úgynevezett *modemek (jelátalakítók)* végzik. A vonal „adó" oldalán levő modem a számítógépes jeleket a hangfrekvenciának megfelelő tartományba alakítja, a „vevő" oldalon levő modem pedig a megérkezett jeleket a számítógép (vagy a terminál) számára visszaalakítja. E folyamatokat nevezzük *adatátvitelnek.*

modemek, adatátvitel

Tudjuk, hogy ma már a tv- és rádióműsorok műholdak segítségével jutnak el a Föld egyik végéből a másikba. Számítógépek is összeköthetők egymással vagy terminálokkal hírközlési műholdakon keresztül.

Olyan igény is van, hogy adatok feldolgozását lehessen távolról elvégeztetni a számítógéppel, vagyis egy, az előzőektől eltérő típusú terminálról azt kell egyrészt közölni a központi számítógéppel, hogy melyik programot vegye elő a háttértárjából, majd ez a program a távoli terminálról beolvassa a feldolgozandó adatokat, s az eredményeket is oda elküldi. Ezért a terminál megfelelő adatbeolvasó ill. nyomtató perifériákkal rendelkezik. Persze egy fordítóprogram lehívásával programfejlesztést is lehet végezni, az időosztásos rendszerekhez hasonlóan ugyanolyan terminállal. Természetesen itt is adatátvitellel oldjuk meg a távolságok áthidalását.

Olyan rendszerek is vannak, melyeknél a központi gépben mindig ugyanaz a program fut, s azzal – általában speciális – terminálokról a program által kezelt nagy adatbáziso(ko)n egy-egy műveletet lehet elvégeztetni. E műveleteket tranzakcióknak nevezik, s magát a rendszert tranzakció-orientáltnak, vagy tranzakciósnak. Ilyen rendszer pl. a repülőgép-helyfoglalás, amelyről a következő fejezetben szólunk részletesebben, vagy a banki, takarékpénztári rendszerek, melyeknek speciális termináljai a fióküzletekben találhatók, s az üzletfelek számláit a betétek, kivétek beküldésével a központi számítógép háttértárolóin tartják naprakészen.

távfeldolgozó hálózatok

Azokat a rendszereket, melyek egy központi számítógépet s adatátvitellel hozzácsatlakoztatott terminálokat tartalmaznak, összefoglaló névvel *távfeldolgozó hálózatok*nak nevezzük.

A személyi számítógépek megjelenésével egyes területeken a távfeldolgozó hálózatoknak „versenytársai" lettek a professzionális személyi számítógépek. Egyedi alkalmazásoknál (és ha a feladat mérete nem lépi túl a professzionális gép korlátait) ez a verseny gyakorlatilag fennáll.

helyi hálózatok

Alig egy pár évvel a professzionális gépek térhódítása után megjelentek a hálózatba kapcsolt személyi számítógépek. Ezek a *lokális vagy helyi hálózatok* általában 4–10 – esetleg több száz – professzionális gépet kapcsolnak össze. A gépek között szokott lenni egy olyan is, melyet *server*-nek, *kiszolgáló*nak neveznek, s melynek célja bővíteni az egyes professzionális gépek korlátait, különösen háttértároló és nyomtatási kapacitás terén. Tipikus alkalmazás pl. a server gép nagy kapacitású winchester tárán elhelyezett adatbázis közös használata.

A számítógép-hálózatokat a számítógépek összekapcsolásának topológiája szerint is osztályozhatjuk. Ha egy központi géphez a többi gép (távfeldolgozó hálózat esetén a terminálok) sugarasan csatlakozik, a hálózatot *csillag felépítésű* hálózatnak nevezzük. Más kapcsolódási formák is elképzelhetők. Így például, ha minden gépet minden géppel összekötünk, *teljesen összekapcsolt* rendszer jön létre. Ez az adat helyétől függetlenül nagyon gyors elérést biztosít, de kiépítése és bővítése nagyon költséges. Kialakítható *gyűrűs felépítésű* rendszer, amelyben a hálózatba kapcsolt gépek önmagában zárt, kétirányú átvitelt biztosító vonalakra vannak „felfűzve". Végül megemlítjük a *hierarchikus felépítésű hálózatot,* amelyben a különböző gépek általában különböző intelligenciát is képviselnek. Nagyon jó példa erre a nagyszámítógépekhez kapcsolt személyi számítógépekből álló rendszer, ha a személyi számítógépek mindegyikéhez még néhány munkahelyet, vagy valamilyen mérési feladatot végző processzort kapcsolnak.

hálózatok felépítése

intelligens terminál

A nagygéphez kapcsolt személyi számítógépet *intelligens terminálnak* is szokás nevezni. Ilyen esetekben a feladatok egy része (például a programírás és -tesztelés) az intelligens terminálon elvégezhető, és csak a nagygépre szánt funkciókat (például a futtatást) küldik el a központi gépre. Ez tehermentesíti a központi gépet, ami végső soron a rendszer átbocsátóképességét javítja.

Végképpen be vagyunk hálózva.

28.

TÚL AZ ÓPERENCIÁN
(Adatátviteli és számítógép-hálózatok)

Túl az óperencián boldogok leszünk – éneklik az operett színpadán, de most nem erről van szó. Hogy mostani mondanivalónkhoz eljussunk, így kell a címet módosítanunk: „Túl az óceánon". Ez ugyan nem annyira költői, de célszerűbben folytatható, s a folytatás sokak számára hihetetlen lesz.

Túl az óceánon, az Egyesült Államok Georgia államának Atlanta nevű városában van egy számítógép. Ennek a számítógépnek néhány terminálja Budapesten van. A gépen üzemel a Gabriel helyfoglalási rendszer, mely számos légitársaság járataira nyilvántartja a helyfoglalásokat. Nyilvánvaló, hogy óriási pazarlás volna, ha mindazon nemzeti légitársaságok irodáinak termináljai, amelyek a helyfoglalási rendszerhez csatlakoznak külön-külön össze lennének kötve – óceánokon keresztül – a szóban forgó géppel. Megoldható, hogy pl. egy ország összes ilyen terminálja egy közös összeköttetést használjon csak, ami nyilván olcsóbb. De mi történik, ha ez az összeköttetés meghibásodik? Ugyanakkor nemcsak ezt az egyetlen gépet, hanem esetleg egy másikat is el kell érni ugyanazon terminálról. Az ilyen nagy kiterjedésű, bonyolult, több számítógépet is tartalmazó rendszerek kommunikációs alapja egy *adatátviteli hálózat,* mely maga is számítógépekből áll, jelen esetben a Légitársaságok Nemzetközi Távközlési Szervezete (SITA) hálózata. E számítógépek, melyek csak az adatok átvitelével, irányításával foglalkoznak, alkotják a hálózat *csomópontjait.* E csomópontokhoz csatlakozhatnak akár a feldolgozó számítógépek, akár a terminálok. Maguk a csomópontok keresztül-kasul vannak összekötve, így biztosítván, hogy valamely összeköttetés meghibásodása esetén egy másik – kerülő – út alkalmazása lehetséges legyen. A terminálokat tehát csak a legközelebbi csomóponthoz kell csatlakoztatni. Mivel a hálózat világméretű, ez aránylag rövid összeköttetéseket igényel csak.

adatátviteli hálózat, csomópontok

Mivel az adatátviteli hálózathoz nemcsak több terminál, hanem több feldolgozó számítógép is csatlakozhat, és a hálózat valamely terminál adatait kérésére akármelyik számítógépre, sőt akármelyik másik terminálra is irányíthatja, *számítógép-hálózatról* beszélünk. Itt a számítógép-hálózatoknak csak két jelentős alkalmazásáról emlékezünk meg. Az egyik az információs adatbankok lekérdezése. Ha pl. kíváncsiak vagyunk arra, hogy az utóbbi félévben melyik folyóiratban és milyen cikkek jelentek meg a személyi számítógépek lokális hálózatainak szabványosításáról, ezt a megfelelő nyugat-európai vagy amerikai adatbankból innen Magyarországról is lekérdezhetjük (olyan terminálról, melynek erre engedélye van, s persze a költségek térítése mellett). A másik alkalmazás az *elektronikus posta.* Levelezhetünk, üzenetet küldhetünk egy másik, a rendszerre rákötött terminálnak. Persze sok problémát kell ilyen nagy rendszerekben megoldani. Néhányat nézzünk meg a Gabriel rendszer szemszögéből.

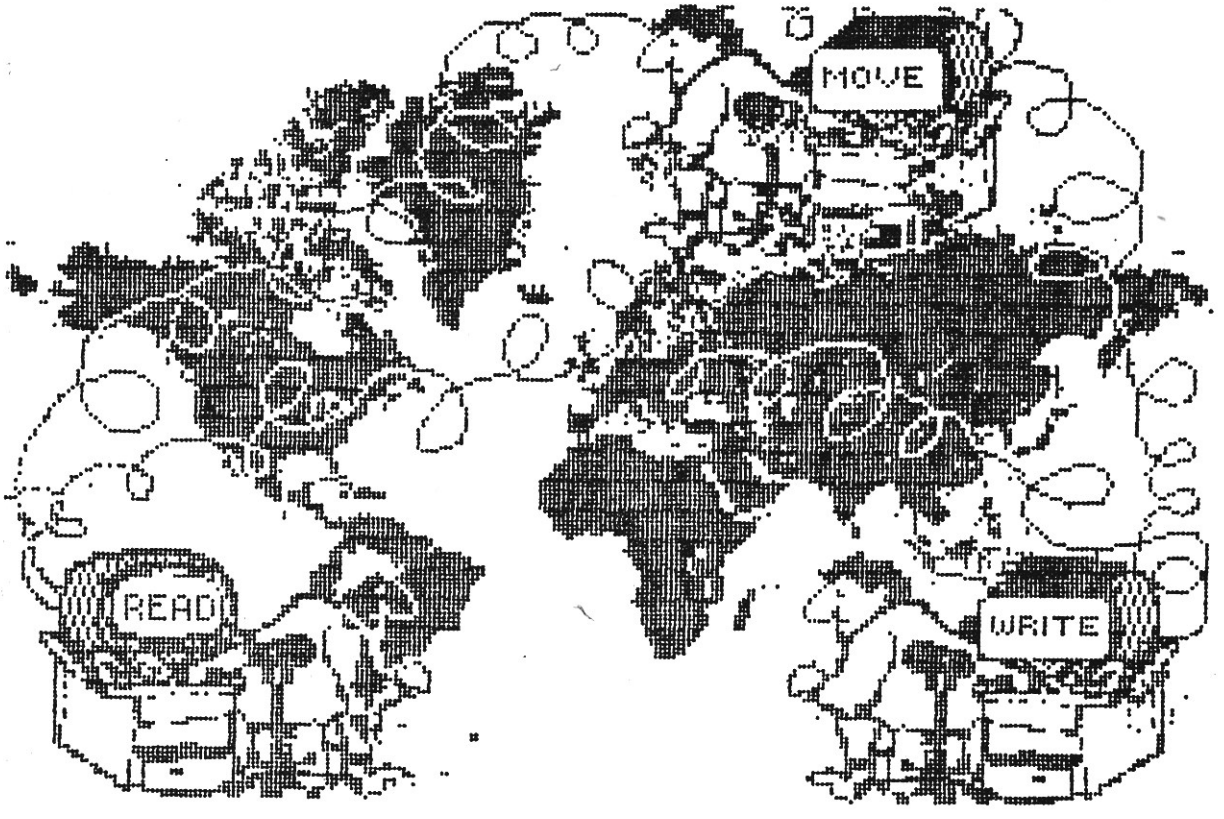
elektronikus posta

A bejelentkezéssel kapcsolatos illetékességi és illetéktelenségi problémákon túl felmerülnek úgynevezett hozzáférési problémák is. Teljesen nyilvánvaló, hogy először is minden légitársaság csak a saját járataival kapcsolatos adattárakhoz férhet hozzá. De ezen belül is vannak megszorítások. Egy-egy jegyiroda termináljáról például csak helyfoglalás végezhető. Ugyanakkor egy-egy ilyen irodában csak azokhoz a korábban végrehajtott

helyfoglalásokhoz lehet hozzáférni, amelyeket ők maguk végeztek. (Nincs tehát mód arra, hogy egy KLM járatra a Malév irodában lefoglalt helyet valaki a KLM irodában módosíthassa.) Biztonsági okok magyarázzák azt is, hogy az utaslistához nem férhet hozzá akárki.

Ezek a jellegzetességek minden más, hasonló elven működő rendszernél megvannak. A bejelentkezni jogosultak ún. felhasználói csoportok valamelyikébe vannak sorolva, s ez határozza meg, hogy mekkora az az adatkör, amelyhez jogosultak hozzáférni.

A bemutatott helyfoglalási szisztéma tipikus példája a tranzakciós rendszereknek. Az ilyen rendszerek jellemzője, hogy egyidejűleg többen dolgozhatnak ugyanannak az adatbázisnak a rekordjaival. A számítógép a bejelentkezéseket sorbaállítja és az érkezés sorrendjében foglalkozik a kérésekkel. Így ha pl. egy járatra már csak egy hely van, ezt az fogja megkapni, aki hamarább (akár csak néhány másodperccel hamarább) tartott rá igényt. A rendszer mindenkor a valóságnak megfelelő helyzetet tartja számon. Egy ilyen tranzakciós rendszerben a felhasználó a gép erőforrásait már nem használhatja szabadon, hanem csak a rendszer céljainak megfelelően. (Többek között ezért nem tudnak a jegyiroda alkalmazottai programokat írogatni termináljaikon ebédidejük alatt.)



adatátviteli hálózatok

Az *adatátviteli hálózatok* megjelenésével a világ összezsugorodott. Ma már másodpercek töredéke alatt lehet akár Ausztráliából vagy Japánból kapcsolatot teremteni Európában lévő számítógépekkel. Nem véletlen, hogy a nagy számítógépgyártók figyelme egyre jobban a műholdas információtovábbítás felé fordul. Az IBM például egyszerűen megvette a Satellite Business System nevű szintén amerikai céget. A nagy európai gyártók (Olivetti, Philips, ICL) közös hírközlési műhold felbocsátását és működtetését tervezik.

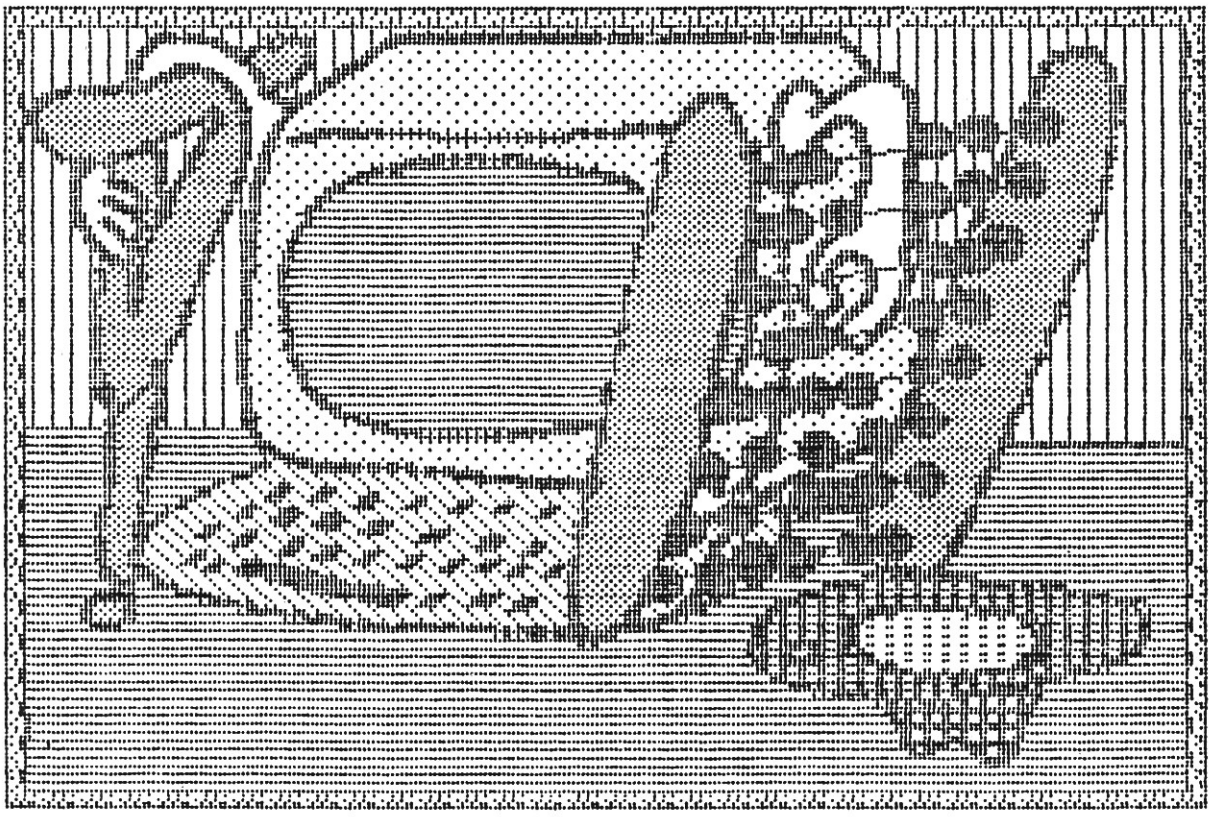
Már csak az a kérdés, hogy ettől boldogabbak leszünk-e az óperencián innen és túl.

29.

SZÁMÍTÁSTECHNIKA A HARDVERES GÖRBE SZEMÉVEL

Hiába minden! A gép csak nem akar működni. Pedig a műszakiak már mindent megpróbáltak! A méregtől és az erőlködéstől veres már a hardveres, és lila ködfoltokon keresztül sárga irigységgel nézi a szoftvereseket, akik gúnyosan mosolyogva szemlélik erőfeszítéseit a gép életrekeltése érdekében. Ők még nem tudják, hogy az első hardverteszt sikeres lefutása után ugyanez a hiba automatikusan átalakul szoftverhibává, – egészen addig, amíg a szoftveresek vissza nem rúgják a gépet, azaz a labdát a hardveresekhez. És ez így folytatódik amíg a gép – esetleg magától – meg nem javul, vagy a felhasználó végül nem hív valakit, aki valóban ért hozzá. Minden kezdő mérnök, még az is, aki egyetlen utóvizsga nélkül végezte el a Műszaki Egyetemet, tudja, hogy *Murphy törvénye* még Ohm törvényénél is erősebb. Ha egy elektronikus készülékben a kettő ellentmondásba kerül egymással, akkor a folyamatok végső eredménye Murphyt, és nem Ohmot fogja igazolni.

Murphy törvénye



Azt azonban már csak a berendezéseket karbantartó mérnökök és technikusok tudják, hogy

- bármilyen szerszám vagy alkatrész, amelyet leejtenek, a műhely vagy a berendezés leghozzáférhetetlenebb sarkába fog beszorulni,
- mielőtt azonban odaér, előbb ráesik a hüvelykujjadra,
- és minél nélkülözhetetlenebb a munka folytatásához, annál nehezebben lehet előkotorni.

Természetesen mindezen sorscsapások ellen védekezhetünk a következő tétel felismerésével és gyakorlati alkalmazásával:

„A padlóról semmi sem eshet le!"

Azzal is tisztában vannak a hardveresek, hogy még a nagy integráltságú áramkörök korában is időnként szükség van vezetékekre, kapcsolókra, csatlakozókra. Ennek a szükségszerű rossznak az egyenes következménye, hogy

— minden amit összeraksz, előbb-utóbb szétesik,
— minden vezeték, noha pontosan méretre levágtad, a készülékbe való beillesztéskor rövidnek bizonyul,
— nem árt, ha üzemeltetés előtt bedugod a hálózati csatlakozót és bekapcsolod a készüléket,
— ha szorul a csatlakozó, erőltesd. Ha eközben összetörik, nem baj, úgy is ki kellett volna cserélni.

Végül egy utolsó jótanács:

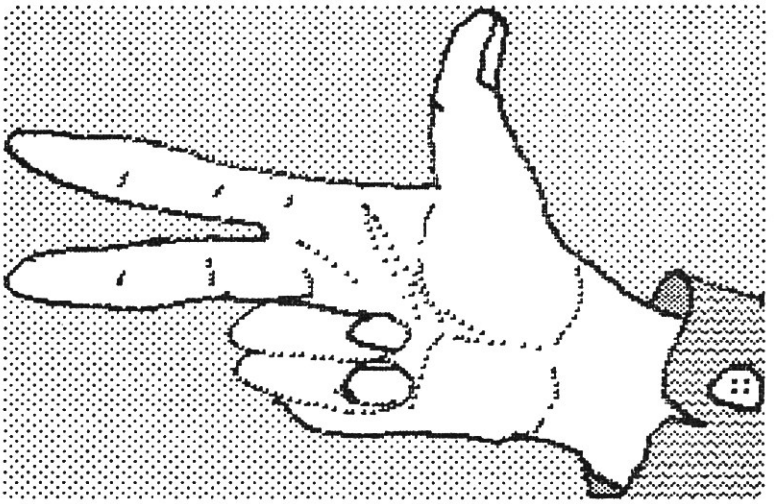
„Ha egy új berendezést semmiképpen sem tudsz üzembe helyezni, akkor próbáld meg elolvasni a kezelési utasítását."

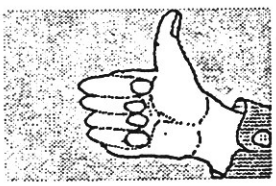
Mindezekre a tanácsokra természetesen csak akkor van szükség, ha a gép szemmel láthatóan nem működik. Tehát még a laikusok, a programozók, a szervezők, sőt még a kis és nagy főnökök is látják, hogy baj van. Egy lámpa sem villog, a lemezegységek nem forognak, a nyomtató nem csattog. Ilyenkor tényleg nincs mese, jó erősen meg kell markolni a csavarhúzót, és nyakig bele kell temetkezni a több lepedőnyi kapcsolási rajzokba. De ha a számítógép csak egy kicsi életjelt is ad, akkor a legjobb, ha a hardveres félórányi lázas látszattevékenység után magabiztosan kijelenti: Szoftverhiba.

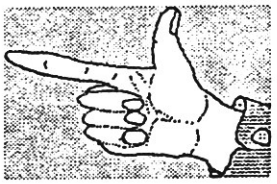
Tíz az egy ellen, hogy ha alaposan utánanéznek, a szoftverben *is* találnak valami hibát.

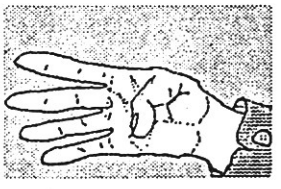
AMITŐL INTELLIGENS:

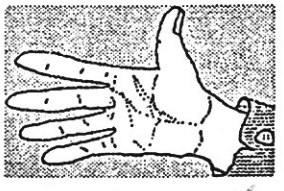
A SZOFTVER







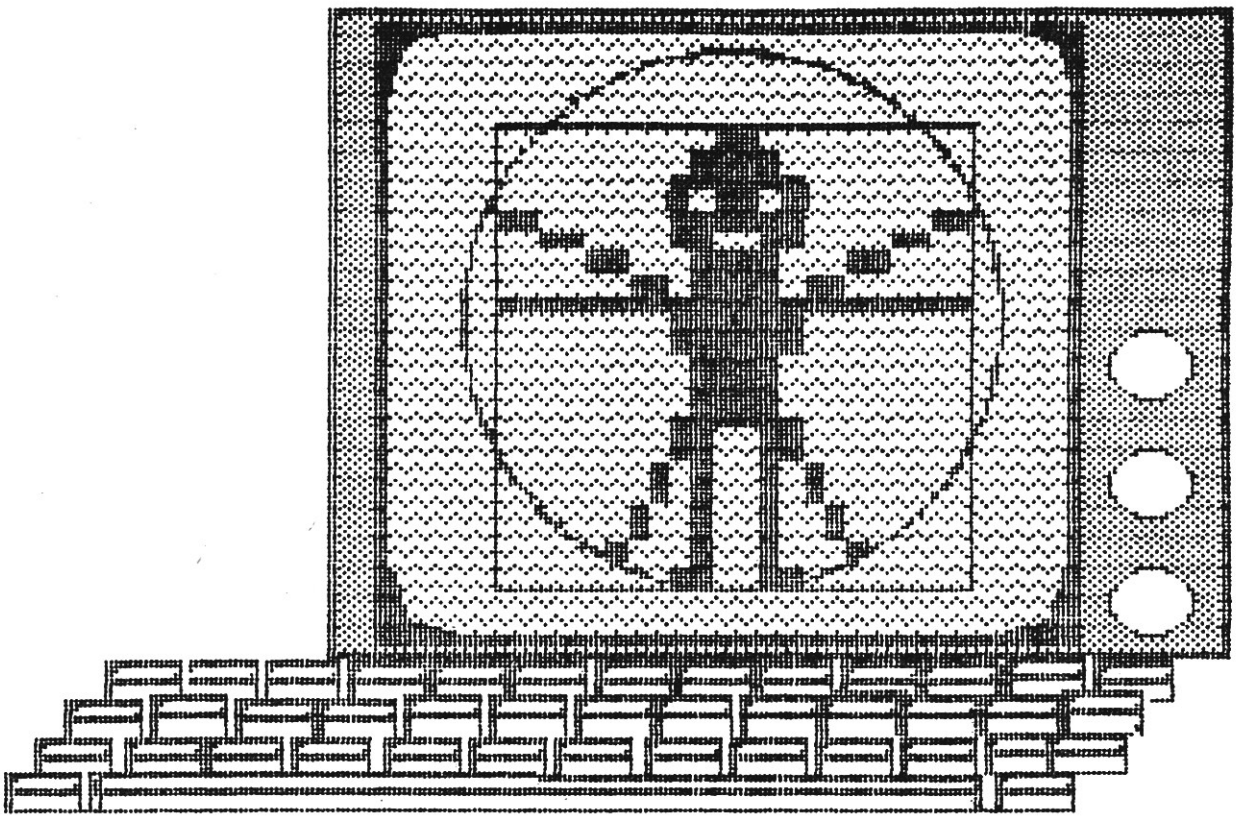




30.

GÉP TESTBEN GÉPEN ÉLEK
(A szoftver legfontosabb összetevői)

Nem valószínű, hogy könyvünk minden olvasójának van saját képmagnója. De ha lenne is, mit érne jó filmek nélkül. Valószínűbb, hogy sok olvasónknak van otthoni számítógépe. Az is valószínű, hogy annak megvétele után mindenkinek első dolga volt egynéhány jó játékprogram beszerzése. Egyszóval a képmagnó önmagában csak szükséges, de nem elégséges feltétele a kellemes időtöltésnek. És ugyanez a helyzet akkor is, ha valaki egy számítógépet ROM-ban levő Basic adta lehetőségeken túl bármilyen más célra alkalmazni akar. (Tudatosan használtuk az „alkalmazni" igét a „használni" helyett. A kettő közötti különbség érzékeltetésére egyetlen példa: aki valamelyik lövöldözős játékban éppen csúcsot javít, az használja, de nem alkalmazza a számítógépet.)



A sikeres alkalmazásokhoz azonban a puszta hardver önmagában nem elegendő. A számítógép hatékony felhasználását a szoftvert alkotó programok biztosítják.

E különböző fontosságú és nagyságú feladatok megoldását a szoftver különféle alkotóelemei látják el. Az elvégzendő feladatok funkciója szerint egy korszerű számítógéphez tartozó szoftver az alábbi módon tagozódik:

- rendszer-szoftver (ezen belül):
  - vezérlőprogramok
  - feldolgozói programok
- alkalmazási szoftver

A rendszer-szoftver legfontosabb feladatai közé tartozik:

- gazdálkodás a hardvereszközökkel,
- a gép működésének felügyelete,
- a háttértárolók és azok adattartalmának karbantartása,
- a programozási munka megkönnyítése,
- ismétlődő feladattípusok megoldásához kész programok vagy programelemek biztosítása,
- a gép minél egyszerűbb és kényelmesebb használatának és alkalmazásának elősegítése.

A felhasználó számára a legközvetlenebb segítséget a feldolgozó programok biztosítják. Ez a segítség elsősorban a programírás megkönnyítését és az állandóan ismétlődő programok, illetve programelemek beépíthetőségét jelenti. A feldolgozó programok között a fordítóprogramok a legfontosabbak. Ezekre azért van szükség, mert minden számítógép alapvetően egyetlen nyelvet ért, egyetlen nyelv utasításait képes közvetlenül végrehajtani: saját anyanyelvét – a *gépi kódot.*

gépi kód

Ráadásul a különböző típusú gépeknek eltérő gépi kódja van. Ezek az egymástól különböző kódok azonban egy dologban mégis azonosak: számjegykombinációk sorozatából állnak, s ezért kellő felkészültség nélkül érthetetlenek. Az ilyen számjegyoszlopok semmit nem árulnak el arról, hogy milyen utasításokat jelentenek. Megértésükhöz alaposan ismerni kell a gép utasításait és működését.

A számítógépek széles körű elterjedését, a hardver kezdetben magas ára mellett éppen ez a bonyolult, az ember köznapi gondolkodásától távol álló és idegen programozhatóságuk gátolta. Olyan közös nyelvre volt szükség, amely az emberi kifejezés és a gépi megértés közötti ellentmondást képes feloldani. Ma már a felhasználó a feladatot megoldó programot a gépi kódhoz képest *magas szintű nyelven* (pl. Fortranban, Cobolban stb.) írja meg. A számítógép-szoftver erőforrásai között vannak olyan *fordítóprogramok,* amelyek működésük során a Fortran, Cobol stb. nyelven írt programot elemzik és gépi kódra fordítják. Az így *lefordított* programot azután – bizonyos kiegészítések elvégzése után – a számítógép már közvetlenül végre tudja hajtani. Az említett ellentmondást, vagyis a közös nyelv hiányát az ember magával a számítógéppel oldja fel. Ez azonban azt jelenti, hogy a magas szintű program előállítása egy külön munkamenet.

fordító-programok

A felhasználó számára más segítséget nyújtanak a vezérlő programok. Ezeket a programokat működtető, vagy *operációs rendszernek* is szokás nevezni.

operációs rendszer, nagy-számítógépek

Nagyszámítógépek esetében az operációs rendszer a számítógép üzemeltetőjének nyújt szolgáltatásokat: például automatikusan regisztrálja a program végrehajtása során felhasznált erőforrásokat s ezek alapján bármikor elkészíthető a gépidő-számla.

A gépkezelő (operátor) az operációs rendszer segítségével bepillanthat a feldolgozás folyamatába: információkat kaphat a gép adott pillanatban végzett tevékenységéről, a hardver pillanatnyi állapotáról, a gépben levő munkák készültségi fokáról. Az operációs rendszer jelzi, ha egy periféria nincs bekapcsolva, ha téves mágnesszalag vagy mágneslemez van feltéve stb. A gépkezelő az operációs rendszerrel az operátori parancsnyelv segítségével érintkezik.

A programozó az általa összeállított munka elvégzéséhez szükséges hardver- és szoftvererőforrás igényt egy másik nyelv, a munkavezérlő nyelv segítségével közli az operációs rendszerrel.

Operációs rendszert használnak a *mikroszámítógépeknél* is. Az otthoni számítógépek egy részénél az ezzel való kommunikáció nehézkes (tulajdonképpen nem is cél, hogy a felhasználó „találkozzon" a rendszerrel). Kényelmes géphasználatot azonban ebben a kategóriában is csak egy könnyen kezelhető operációs rendszer tud nyújtani. A professzionális

mikrogépek operációs rendszere ezt a kényelmet sok területen megadja. Így például könnyen lehet informálódni a meglevő programok nevéről, szükség esetén egyszerűen változtatható egy program neve, egyetlen paranccsal törölhető a program, ha már nincs rá szükség. Ugyancsak egyszerűen kiírható a képernyőre a szükséges program, vagy készíthető róla másolat. Az operációs rendszer biztosítja, hogy a program létrehozása, javítása, bővítése, egyes utasításainak törlése, egyszóval a programszerkesztés nagyon egyszerű módon megvalósítható legyen. Mivel a mikroszámítógépekhez lényegesen könnyebben és egyszerűbben lehet hozzájutni, mint a nagyszámítógépekhez, fontos, hogy a számítástechnikailag képzetlenebb felhasználó is könnyen tudjon dolgozni. Ezt a jó operációs rendszer segíti elő.

A következő fejezetekben a szoftverhez kapcsolódó fogalmakkal, problémákkal ismerkedünk meg.

31.

HOGYAN LESZ A BOLHÁBÓL ELEFÁNT?
(Operációs rendszerek)

Ha egy professzionális személyi számítógépet a megvásárlás után kicsomagolunk és bekapcsolunk, azt tapasztaljuk, hogy a gép általában nem működőképes.

Egy otthoni számítógép esetében ugyanilyen körülmények között bejelentkezik a Basic ↄ majd a READY kiírással jelzi, hogy kész a felhasználó kiszolgálására. Azt jelentené mindez, hogy az „alacsonyabb" kategóriába tartozó gép intelligensebb, mint a profi testvére?

Korántsem. Arról van szó, hogy egy számítógép hardver akkor válik működőképessé a felhasználó számára, ha már él a géphez tartozó működtető (operációs) rendszer. Mindaddig, amíg ez nem következik be, a gép csak annyit tud, amennyit a hardver és a ROM-ban elhelyezett (hardveresített) programok képesek megvalósítani.

Az otthoni számítógép kategóriájába tartozó gépeknél a Basic a ROM-ban van, és a gép bekapcsolását követően automatikusan aktivizálódik.

A professzionális gépek ennél bonyolultabb és fejlettebb működtető rendszerrel rendelkeznek. Ezt a rendszert természetesen a géppel együtt adják. Mivel ez a program már nagyobb méretű, és a felhasználónak meg akarják adni a lehetőséget, hogy igényei szerint, illetve a gép kiépítettségének megfelelően „fazonírozza", így az operációs rendszer nem kerül a ROM-ba, betöltéséről tehát gondoskodni kell. (Egyes professzionális gépek szintén használnak ROM-ba égetett Basic interpretert, de ez nem jellemző.)

A hagyományos számítógépeknél az operátornak minden bekapcsolás után rendszert kell indítania. A rendszerindítás során az operátor az operációs rendszert (illetve annak egy részét, az ún. rezidenst) mágneslemezről a tárba tölti. A betöltést természetesen program végzi, az operátor ezt a programot indítja el.

A professzionális gépek esetében — a kényelmes használhatóság érdekében — a gép rendszerbeállítása során kell létrehozni a „testreszabott" változatot, vagy változtatás nélkül használatba venni a géppel kapott működtető rendszert.

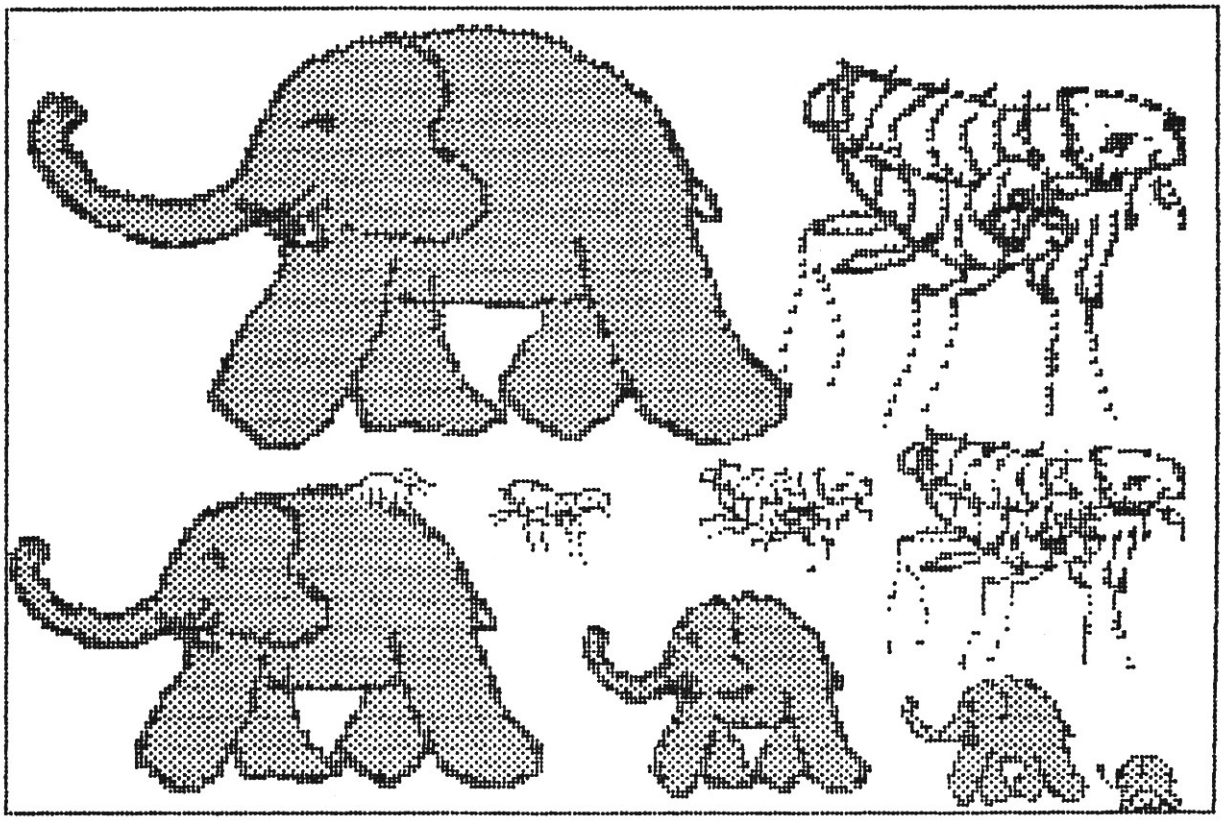
A winchesteres gépek esetén ez a használatba vétel azt jelenti, hogy a hajlékonylemezen kapott rendszert — az esetleges módosításokkal — fel kell tölteni a fixlemezes háttértárra. Winchester nélküli gép esetében az egyik meghajtóba kell helyezni a rendszert tartalmazó hajlékonylemezt.

Mindkét esetben a bekapcsolást követően a ROM-ból automatikusan elindul egy olyan rövid program, amely a működtető rendszert betölti. (Itt tehát az operátor tevékenysége automatizálva van.)

Az operációs rendszer a legfontosabb szoftverkomponens, amely a számítógép működését vezérli. Felhasználói parancsokat olvas be, s annak hatására végez el, vagy végeztet el valamilyen tevékenységet. Funkciója hasonlít egy vezető diszpécseréhez, aki a kapott feladatokat kiosztja a gyáregység munkásai között, ellenőrzi azok végrehajtását, általában törődik a kapacitások kihasználtságával, a hibákkal stb.

Az operációs rendszerek az első generációs gépeknél teljesen hiányoztak. A második generációnál már megjelentek az egyszerű, egy felhasználót kiszolgáló változataik. Az összetett, távfeldolgozást, hálózatba kapcsolást is megengedő változatok a harmadik és negyedik generációra jellemzőek.

Az operációs rendszer kiszélesíti egy számítógép lehetőségeit, kényelmessé teszi annak kezelését, használatát. Minél jobb, fejlettebb, annál jobbnak érzi a felhasználó (esetleg ugyanazt) a gépet.



A felhasználó és gép kapcsolatának módja szerint lehetnek

- kötegelt (batch) feldolgozást biztosító,
- párbeszédes (interaktív) feldolgozást támogató operációs rendszerek.

kötegelt feldolgozás

A *kötegelt feldolgozás* során általában a gép a felhasználó parancsait nem közvetlenül a felhasználótól – konzolról vagy terminálról –, hanem egy előzőleg gondosan előkészített, a gépbe betáplált munkavezérlő programból (job control program) kapja. A felhasználó nincs közvetlen kapcsolatban a számítógéppel, a munkavezérlő parancsait általában leadja egy üzemeltető szolgálatnak, s az táplálja be a számítógépbe. A gép sorra veszi e parancsokat, és végrehajtja, amit ezek előírnak. Az eredményekért a felhasználó egy-két óra (esetleg nap) múlva jelentkezhet. A 33. fejezetben látni fogjuk, hogy mindez hogyan valósulhat meg.

interaktív rendszer

Az *interaktív* (párbeszédes) rendszerben a felhasználó a parancsokat közvetlenül, egyenként adja meg a gépnek. A gép a végrehajtás után azonnal visszajelez, így állandó párbeszéd zajlik a gép és a felhasználó között.

A professzionális személyi számítógépeknél a parancsok egy része belső parancs, ezeket a rendszer azonnal megérti. Más részük külső parancs. A külső parancs nem más, mint egy végrehajtható programnak a neve. A nagyszámítógépeknél az ismétlődő parancsokat el lehet tenni egy erre a célra szolgáló könyvtárba, ahonnan szükség esetén egyetlen utasí-

tással akár száz parancsot is elő lehet hívni és végre lehet hajtatni. Az állomány- vagy adatkezelő rendszer az operációs rendszerek másik fontos komponense. (Az operációs rendszerek parancsairól az 50. fejezetben lesz szó.)

A személyi számítógépek esetén az operációs rendszerrel a billentyűzeten keresztül lehet a kapcsolatot tartani.

Nagyszámítógépeknél a felhasználó és az operációs rendszer közötti kapcsolat nyelvét munkavezérlő nyelvnek nevezik. A kötegelt feldolgozás esetén (de sok időosztásos rendszernél is) a felhasználó elveszíti a kapcsolatát a programmal annak futása során. Ezért az összes parancsot a programmal és az adatokkal együtt előre össze kell állítani. Az így keletkező utasításcsomag (a régebbi gépeknél általában egy kártyaköteg) a gép számára egy munkát (job-ot) jelent. Az operációs rendszer ezt elemzi, és innen veszi a végrehajtáshoz szükséges információkat.

Az operációs rendszer teszi tehát kényelmessé, hatékonnyá, hatalmassá a számítógépet. Ezen a tényen alapszik az a találós kérdés, amely a válasszal együtt így szól:

Mi az elefánt?

„Az elefánt egy olyan bolha, amely hatékony operációs rendszerrel van felszerelve."

32.

HÉT NYELVEN BESZÉL
(Programozási nyelvek)

Egy alkalommal megkérdeztem az egyik programozótól, hány nyelvet ismer. Hetet, válaszolta: a Basic, Cobol, PL1, Fortran, Pascal, Assembler és a magyar nyelvet. Ezekkel a nagyvilágban nemigen számítana műveltnek, de a számítógépek világában könnyen mozoghat. A felsoroltak közül az első hat nyelv olyan, amelynek segítségével a számítógépeknek lehet utasításokat adni.

Basic

Mint azt már írtuk, a számítógép hardverje csak az 1 és 0 jelek sorozatából álló utasításokat, az ún. gépi kódot tudja értelmezni. Ebből állapítja meg, milyen műveletet kell elvégezni, hol vannak a memóriában az ehhez szükséges adatok, és hová kell az eredményeket elhelyezni. Egy program megírása gépi kódban azonban igen nehéz és igen hosszú lenne. Egy összetett matematikai kifejezés kiszámolásánál minden egyes műveletet külön kellene kódolni és például egy 1 Mbájtos tárolójú gépnél, ha közvetlenül kellene megadni az adatok címét, akkor ezek mindegyikénél egy 20 jegyű bináris számot kellene kiírnunk. A programozási munkának a megkönnyítésére alkották meg a különböző programnyelveket. Ezeknek a nyelveknek nagyon szigorú szabályai vannak. Nem tűrik a kétértelmű kifejezéseket, de a helyesírási hibákat sem. Ám elég egyszerűek ahhoz, hogy az ember – gyakorlattól függően – pár naptól néhány hónapig terjedő idő alatt megtanulhassa használatukat. Ugyanakkor eléggé össze tudják vonni a gépi utasításokat ahhoz, hogy a

programozók ne ezek formájára, hanem a végrehajtandó eljárások logikájára összpontosíthassanak. Így például a legtöbb programozási nyelvben több műveletet is végre lehet hajtani egy utasítás segítségével és a benne szereplő adatok tárolóbeli helyét sem kell pontosan ismernünk, elég, ha egy névvel hivatkozunk rájuk.

Az így megírt programokat hiába olvassuk be a gépbe, az még nem hogy nem tudja az utasításait végrehajtani, de még csak nem is érti, mit kellene tennie. Mivel azonban a programozási nyelvek szabályai egyértelműek, lehet olyan számítógépes programokat készíteni, melyek az előbbi módon és nyelven megírt programokat átírják a gép számára érthető gépi kódba. Ezeket hívják *fordítóprogramoknak.* A fordítóprogram bemenete a programnyelven megírt *forrásprogram,* végső eredménye pedig a gép által közvetlenül értelmezhető gépi kódú program. Ezt az átalakítást *fordításnak* nevezzük. Ennek lényege az, hogy a nyelv szavakkal megfogalmazott utasításait és az azokban névvel megadott operandusokat a fordítóprogram lefordítja bináris formájú gépi utasításokra és címekre.

fordító-program, forrás-program, fordítás

E művelet közben a fordítóprogram bizonyos fokú ellenőrzést is végez és ha valamit nem tud értelmezni a forrásprogramból, akkor azt jelzi. Természetesen attól, hogy hibátlannak talált egy programot, még nem biztos, hogy az valóban jó, és valóban azt csinálja, amit készítője gondolt. A tartalom helyességének vizsgálatára ugyanis még a legjobb fordítóprogram sem képes.

A magas szintű programozási nyelvek előnye még az is, hogyha ugyanazt a feladatot más típusú gépen akarjuk végrehajtani, akkor azt nem kell újból programozni. Az egyszer megírt forrásprogramot a másik gép fordítóprogramja a neki megfelelő gépi kódra fordítja le. (Valójában a gyakorlatban nem ilyen szép a helyzet, a legtöbb gép a szabványos nyelvet valamilyen kis „tájszólással beszéli". Az abból adódó eltérések kijavítása azonban lényegesen kisebb munka, mint a program újraírása.)

Személyi számítógépeknél a leggyakrabban használt programozási nyelv egyszerűsége, könnyen megtanulható volta miatt a Basic. Nagyobb gépeknél az adatfeldolgozó programok többségét Cobolban vagy PL1-ben, a matematikai feladatokat megoldókat pedig általában PL-1ben, Pascalban vagy Fortranban készítik el. A bevezetőben említett művelt programozó által ismert nyelveken kívül még igen sok egyéb általános, vagy speciális feladatokra készült programnyelv van. Szerencsére a számítógépeket nem zavarja meg ez a bábeli nyelvzavar. Amelyikhez van fordítóprogramjuk, azt megértik, amelyikhez nincsen, annak utasításait még figyelembe venni sem képesek, mutagathatunk neki kézzel-lábbal.

33.

KÖZÖS GÉPNEK NEM TÚROS A HÁTA
(Multiprogramozás)

Ahhoz, hogy a számítógép hasznos munkát tudjon végezni, adatokra van szüksége. Egy közepes teljesítményű számítógép egy ötjegyű szám bevitele alatt még akkor is több százezer műveletet képes elvégezni, ha a billentyűzeten igen gyakorlott gépkezelő dolgozik, feltéve, hogy a gépnek van mivel dolgoznia. Ha ugyanezt a számot eredményként írnánk ki akár a leggyorsabb nyomtatón, még így is legalább 100 műveletet tudna végrehajtani a gép a nyomtatás ideje alatt. Valamivel kedvezőbb a helyzet akkor, ha az adatok számítógéphez kapcsolódó háttértárolón, mágnesszalagon vagy mágneslemezen vannak. Ilyenkor a beolvasás és a kiírás „csupán" egy-két nagyságrenddel lassabb a gép műveleti sebességénél, feltéve, hogy a használni kívánt periférián az író-olvasó fej éppen ott áll, ahonnan az adatokat olvasni, vagy ahová írni akarjuk. Ez azonban csak ritkán fordul elő, rendszerint várni kell még a fejnek a megfelelő helyre való beállására is, hogy a fej alá a megfelelő szalagrész vagy lemezfelület kerüljön. Ha ez ellen nem tennénk semmit, akkor a feldolgozás és az adatok ki- és bevitele közti óriási sebességkülönbség következtében a számítógép egyik legdrágább része, a központi vezérlő egység, az idő legnagyobb részében tétlenül állna. Várná, hogy megkapjon egy adatot, vagy várna azért, mert ki kell írja az eredményt. Ezenkívül az operatív memóriát sem tudnánk sokszor megfelelően kihasználni.

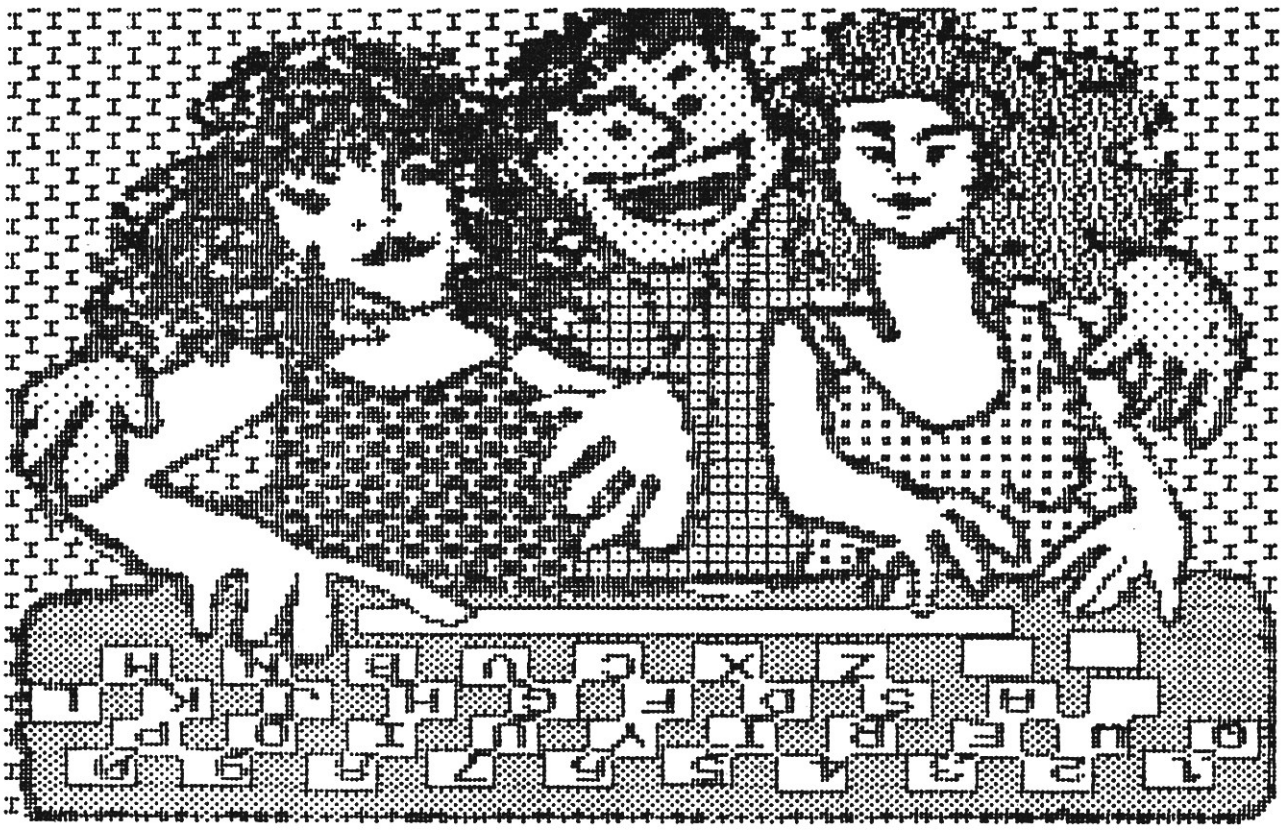
Rövid programok esetén a tár nagy része üresen marad, de ugyanakkor egyes nagy programok miatt mégis szükségünk van a nagyobb tárra. Ez a tétlenség és pazarlás olcsóbb személyi számítógépeknél, ahol egy ki-beviteli egység ára összemérhető a processzoréval és az operatív memóriával, ugyanúgy elintézhető, mint ahogy a kis fizetésű dolgozó morfondírozik: ezért a pénzért nem várhatják el, hogy állandóan dolgozzak, bár az álláspontja azért erkölcsileg elítélendő. Nagyobb teljesítményű gépeknél, és ebből a szempontból ide tartozik a drágább miniszámítógépek egy része is, ezt a fényűzést már nem engedhetjük meg. Ezeknél a központi egység egy nagyságrenddel is többe kerülhet, mint a szokványos perifériák. Az erkölcsi elítélés mellett tényleges intézkedésekre is szükség van. Éppen ezért érthető volt a számítógéptervezőknek és -gyártóknak az a törekvése, hogy olyan berendezéseket készítsenek, melyek lehetővé teszik a központi egység jobb kihasználását, még úgy is, hogy az emiatt valamivel bonyolultabb és drágább lesz.

idő-osztásos ütemezés

Erre a célra fejlesztették ki a multiprogramozású számítógépeket. Ezeknek tárolójában egyidejűleg több program helyezkedik el, melyeken a processzor egyszerre tud dolgozni. A multiprogramozású gépek közül a legelterjedtebbek közé tartoznak az *időosztásos* rendszerek. Minden program kap egy rövid időszeletet a processzortól. Ezen idő alatt a vezérlő és az aritmetikai logikai egység csak ennek a programnak az utasításait hajtja végre. Az időszelet befejeztével félbeszakítják ezt a programot, megjegyzik, hol hagyták abba, és rátérnek a következő – korábban már elkezdett – program utasításaira, – szintén egy időszelet erejéig. A különböző programok futásának ez az egyidejűsége ugyanúgy csak látszólagos, mint a filmeken látható folyamatos mozgás, mely egymás után igen gyorsan levetített állóképekből tevődik össze.

Valójában egy adott időpillanatban a gépben is csak egy program fut. A programok azonban olyan gyorsan váltják egymást, hogy a felhasználó számára a saját programjának felfüggesztései összemosódnak a tényleges feldolgozással és úgy érzékeli, mintha a gép csak neki dolgozna. Az adatok be- és kivitelét az egyes programok számára a központi processzortól független, annál jóval egyszerűbb és olcsóbb, csupán erre a célra szolgáló önálló be/kiviteli processzorok végzik el azalatt az idő alatt, amíg a központi processzor a többi programon dolgozik. Olyankor, amikor egy program adatait egy be/kiviteli processzoron viszik át a memória és a perifériák között, akkor az a reá jutó időszeletet átadja a soron következő programnak, hiszen a processzornak meg kellene várni az adatok beolvasásának illetve kiírásának befejeződését.

Az időosztásos módszer előnye, hogy minden program egyenletes ütemben szóhoz illetve processzorhoz jut. Ezért a sok terminálon keresztül hozzáférhető rendszerek általában ilyen elven osztják szét az egyes programok között a processzor időt. Ez az egyenlősdiség azonban egyben hátrány is. A tisztán időosztásos rendszerben nincsen lehetőség a fontos feladatok előbbrehozására. Jelenlegi gazdasági életünk fő problémája itt is érvényesül — mindenkinek jut valami, de a legjobbaknak ez lehet, hogy kevés. Ezen a gondon segít egy másik multiprogramozási mechanizmus, a *prioritásos* (elsőbbségen alapuló) *ütemezés,* melyet a következő fejezetben ismertetünk.



Mindkét fajta ütemezési elvnél lényeges, hogy a különböző programok ne zavarják egymást, hogy mindegyik tudja, mikor és milyen utasítástól kezdődően kerül sorra, a központi tárolónak mely részeit és milyen perifériákat használhat. Ehhez egy külön vezérlő programrendszerre van szükség, mely mindezeket számon tartja és irányítja. Ezt a számítógépet gyártó cég a géppel együtt szállítja, és ennek hatékonysága legalább annyira befolyásolja a gép tényleges teljesítményét, mint a közvetlen műszaki adottságok. Igaz, hogy ennek a vezérlőrendszernek a működtetése értékes időt és helyet vesz el a számítógépet felhasználni kívánók programjaitól, de sokkal többet nyerünk révén a réven, mint amit el-

vesztünk a vámon. A központi egység lényegesen jobb kihasználásával megnövekszik a számítógép teljesítőképessége, ugyanazon idő alatt jóval több feladatot tud megoldani.

Egyszerűbb vezérlőprogram esetén, ha egy program befejeződik, akkor a gépkezelőnek kell a helyébe egy másikat bevinni, és azt elindítani. Az ember reakciója és a gép tevékenysége közti óriási időeltérések miatt ez akkor is még sok mindenre felhasználható gépidő kiesését jelentené, ha a gép kezelője állandóan feszülten figyelné, mikor kell ebből a célból beavatkoznia és azonnal, hiba nélkül cselekedne. De a gép kezelője nem gép — hanem ember! Van amikor nem figyel, néha lassan cselekszik, sőt — Uram bocsá'! —, esetleg még hibázhat is a soron következő program kiválasztásában és elindításában.

Ezért a fejlettebb vezérlőprogramok ezt a tevékenységet is átvállalják a gép kezelőjétől. Egy program befejezése után automatikusan elindítják a soron következőt, intézkednek arról, hogy az ehhez szükséges adathordozókat az operátorok a megfelelő perifériákra föltegyék. A legfejlettebb vezérlőrendszerek pedig még azt is meghatározzák, milyen működési rend esetén a leghatékonyabb a gép kihasználása. A gépkezelő bejuttatja a gépbe az összes elvégzendő feladatot és az azok végrehajtásához szükséges feltételek pontos leírását. Ezekből aztán a vezérlőprogram saját maga állítja össze az elvégzendő munkafolyamatokat úgy, hogy a feldolgozás folyamán a gép minden része a lehető legjobban legyen kihasználva. Ugyancsak a vezérlőprogram ügyel arra, hogy minden program csak a „saját portája előtt söpörjön". Hiába ücsörögnek békés társbérletben egymás mellett az operatív memóriában, és hiába versengenek a processzor kegyeiért, a vezérlőprogram mint szigorú ellenőr figyeli, hogy minden program csak azokat a perifériákat és tárolórészeket használhassa, amelyeket a számára kijelölt.

34.

PROTEKCIÓSOK ELŐRE!
(A multiprogramozás ütemezése)

Az előző fejezetben ismertettük a multiprogramozású gépek egyik lehetséges ütemezési eljárását, az időosztásos módszert. Ebben a hazánkban oly sokszor kifogásolt, mégis oly gyakori eloszlási elv, az egyenlősdi érvényesül. Mindegyik program ugyanakkora időszeletet, általában legföljebb 1–3 ezred másodpercet kap egy-egy alkalommal. Ha ezt elhasználta, hiába volna még igen sok fontos teendője, ahhoz, hogy ismét a prcesszorhoz juthasson, ki kell várnia, amíg a vele párhuzamosan futó összes többi program is ledogozza a saját idejét. Ez csak akkor rövidülhet meg valamelyest, ha egy program nem tudja végig kihasználni a saját időszeletét, mert például várnia kell egy adatra, adatcsoportra, egy beviteli-kiviteli műveletnek a befejezésére. Ebben az esetben a processzor nem foglalkozik tovább ezzel a programmal, hanem áttér a soron következő program utasításainak végrehajtására.

Az egyenlősdi mellett azonban divatban van nálunk egy másik elosztási mód, a protekciós is. A multiprogramozású gépek ezt is jól ismerik, körükben ezt legalább olyan gyakran megtaláljuk, mint a való életben. Ennél a módszernél a programokat valamilyen elv szerint rangsorolják. A legfontosabb lesz a legmagasabb prioritású, a következő a második legnagyobb prioritású, és így tovább. A gép először mindig a legmagasabb prioritású program utasításait kezdi végrehajtani. Ezt csinálja egészen addig, amíg ezt folyamatosan meg tudja tenni. Mihelyt azonban valamilyen okból nem tud tovább haladni (például várnia kell egy adat beolvasására), a gép (pontosabban a vezérlő program) átadja ezt a feladatot a beviteli-kiviteli processzornak, megjegyzi, hol hagyta abba ezt a programot, majd rátér a következő legmagasabb prioritású program soron következő utasításaira. Ezen addig dolgozik, amíg ismért várakozásra kényszerül. Ekkor a prioritásban ezután következő programra tér át. Amikor egy ennél magasabb prioritású program által működésbe hozott beviteli-kiviteli processzor jelzi, hogy befejezte az adatátvitelt, azaz hogy egy magasabb prioritású programnak már a rendelkezésére áll a kívánt adat, akkor a processzor ismét ezen fog továbbdolgozni. A *prioritásos elven* alapuló multiprogramozásnál mindig a legmagasabb prioritású, munkára készen álló programé az elsőbbség, azaz a processzor. A többi program, hiába áll minden készen a futásához, sorban, türelmesen várakozik.

a prioritás elve

A dolognak kényes pontja a prioritási sorrend megállapítása. Ez történhet azon az alapon, hogy kinek számára szolgáltat a program eredményeket (nyílt protekció), mennyire sürgősek az eredmények (burkolt protekció, mert ez általában összefüggésben van azzal, hogy ki várja azokat) és úgy is, hogy a gép adott idő alatt a lehető legtöbb feladatot tudja elvégezni (ésszerű, vagy igazságos protekciózás). Ez utóbbi esetben azok a programok kapnak magas prioritást, melyek gyakran használnak perifériákat, mert ilyenkor ezek automatikusan szóhoz, illetve processzorhoz engednek más programokat is. Azok a programok viszont, amelyek intenzíven használják a processzort, erősen számolásigényesek, alacsony prioritású besorolást kapnak, hisz maguktól nemigen mondanának le a processzorról, csak úgy, ha azt egy magasabb prioritású program elveszi tőlük. Korszerű, nagyon

nagy teljesítményű gépeknél a szubjektivitás kiküszöbölése és a hatékonyság növelése érdekében rendszerint a gép saját maga is figyeli a prioritási sort, hogy mennyire „önzőek" a nagy prioritású programok. Ezek prioritását, ha nem mondanak le néha maguktól a processzorról, időnként automatikusan csökkenti, míg a „szerény", a processzort másoknak önként gyakran átengedő programok prioritását megnöveli.

Így (is) lesznek az utolsókból néha elsők. Ennyiben fejlettebb a számítógép, mint a társadalom.

35.

MINDENKINEK SZÜKSÉGLETE SZERINT
(Memóriagazdálkodás)

Multiprogramozású gépeknél – a nagyobb számítógépek mind ilyenek – szabályozni kell, hogy az egyszerre futó programok közül melyik foglalhatja le magának a processzort és a központi tárat. Mint azt az előző fejezetekben bemutattuk, a processzort az egyes programok az egyenlősdi (időosztás) vagy a protekció (prioritás) vagy esetleg ezek valamilyen kombinációja (pl. azonos prioritású programok között időosztás) alapján vehetik igénybe. A korszerű számítógépek azonban a központi tár egyidejű használatát is lehetővé teszik. Ezáltal az operatív memóriát minden program szükséglete szerint használhatja, – legalább is addig, amíg a készlet tart, azaz amíg a tár nagysága ezt lehetővé teszi.

Az operatív memória azonban nem lehet akármekkora. Méretét műszaki, de főleg gazdasági okok korlátozzák. Ugyanakkor egy bonyolult feladatot elvégző programnak gyakran több százezer bájtnyi memóriára is szükség lehet feladata ellátásához. Ha a rendelkezésre álló központi tár kapacitása nem elég ehhez, úgy a programozónak saját magának kell gondoskodni arról, hogy programjának azon részei, amelyek éppen nem aktívak (azaz azok az utasításcsoportok, amelyeket az adott pillanatban és a közeljövőben előreláthatólag nem kell majd végrehajtani), ideiglenesen kikerüljenek mágneslemezre és csak akkor jöjjenek be az operatív tárba, amikor éppen szükség van rájuk. Ilyenkor természetesen gondoskodnia kell arról is, hogy helyettük olyan programrészek kerüljenek ki a központi tárból, melyekre éppen akkor nincsen szükség. A programrészeknek ez az állandó ki- és begurigázása a korlátozott méretű operatív memória és a jóval nagyobb kapacitású külső tároló, általában mágneslemez között igen nagy mértékben megnehezíti a programozó munkáját. Az ilyen programokat „átlapolásos"-nak (overlay) nevezzük.

„virtuális" tár, partíciók

A legkorszerűbb gépeknél ezt a feladatot automatikusan elvégzi a számítógép. A programozók úgy írják a programjaikat, mint hogyha azok egy a valóságosnál sokkal nagyobb úgynevezett *„virtuális" tárban* helyezkednének el. A virtuális tár egy része (a kisebbik része) megegyezik a valóságos operatív memóriával, nagyobbik része egy erre fenntartott mágneslemez terület. Ebből a virtuális tárból (látszólagos operatív memóriából) a valóságos tárba a rendszer hívja be mindig azokat a programrészeket, melyek a végrehajtandó utasításokat illetve az azokhoz szükséges adatokat tartalmazzák. Míg az operatív memóriában rendszerint 64 kbájttól (személyi számítógép) általában 1–4 Mbájtig terjed a programok által lefoglalható tártartomány (melynek egy részét a számítógép operációs rendszere használja), addig a virtuális tár mérete ugyanilyen kategóriájú gépeknél elérheti a 16 Mbájtot is, sőt még ennél is jóval nagyobb lehet.

A tényleges illetve a virtuális tár méretén belül minden egyes program igényei szerint kapja meg a futásához szükséges méretű tárat. Ez történhet úgy, hogy a tár eleve több, különféle nagyságú részre, *partíciókra* van osztva. A partícióknak a száma és a mérete is rögzített. Annak ellenére, hogy ezeknek a partícióknak a nagyságát a gépkezelő (amennyiben szükség van rá) időnként megváltoztathatja, ez a rögzített partíciójú tárfelosztás nem elég gazdaságos. Ha például három darab, egyenként 200 kbájtos partíciónk van, melyek mindegyikében egy-egy 100 kbájtos program helyezkedik el, úgy mindegyik par-

tícióban 100, azaz összesen 300 kbájt kihasználatlan marad. Hiába akarnánk azonban ebben egy újabb, akár csak 50 kbájtos programot elhelyezni, erre nincs lehetőségünk. A partíciók merev határai nem teszik lehetővé sem a partíción belüli szabad helyeknek más programok számára való felhasználását, sem az összevonását.

Jobb tárkihasználást tesz lehetővé a változó nagyságú és számú partíciókra való felosztás. Ebben az esetben a programok tényleges nagysága szabja meg az egyes partíciók méretét. Egészen addig, amíg még van elegendő nagyságú szabad tárolóhely, lehetséges további programokat bevinni a memóriába, hogy a multiprogramozás következtében megnövekedjék a gép által elvégezhető munka mennyisége.

36.

CSAK AKKOR MAGAD, HA SZOLGÁD NINCSEN
(Segédprogramok)

A számítógépek használata során nagyon sok olyan feladat, probléma adódik, amelynek megoldása valamilyen programmal lehetséges volna, ha lenne ilyen. Egy program viszont úgy születik, hogy azt valaki valamikor megírja, teszteli, használhatónak ítéli. Ez a valaki nem okvetlenül a felhasználó, vagyis mi magunk, hanem esetleg a gépet gyártó cég szoftverrészlege, esetleg valaki más. Vannak olyan problémák, amelyek egyrészt meglehetősen gyakran jelentkeznek egy számítógép működtetése során, másrészt nem tartoznak sem a szorosabb értelemben vett operációs rendszer, sem a fordítóprogramok „hatáskörébe". A meglehetősen gyakori használati igény miatt jó, ha ezeket a programokat nem a felhasználónak kell elkészítenie.

Az ilyen célt szolgáló programokat nevezzük *segédprogramoknak,* vagy szolgáltatóprogramoknak (eredeti angol elnevezésük szerint utility programoknak).

A többfelhasználós nagyszámítógépeknél ezek a programok elsősorban az üzemeltető számára jelentenek segítséget, de sok olyan szolgáltatásuk van, amelyet a felhasználó is igénybe vesz. Látni fogjuk, hogy több olyan tevékenység is van, amely egy nagygépes környezetben az üzemeltető, esetleg a rendszerszoftveresek feladata, míg az egyfelhasználós személyi számítógépeknél — éppen a könnyű, gyors, kényelmes használhatóság érdekében — az operációs rendszer komponenseként használható.

A teljesség igénye nélkül bemutatunk néhány olyan feladatot, amely a nagyszámítógépeknél segédprogramok segítségével oldható meg.

Mágnesszalagokra és mágneslemezekre való adatfelvitel előtt gondoskodni kell arról, hogy a szalag-, illetve a lemezkötegek elejére felkerüljenek a rendszer számára szükséges információk, a lemezen kialakuljanak a sávok. Mágnesszalagok címkézését és mágneslemezek inicializálását e célra szolgáló segédprogramokkal lehet elvégeztetni.

Ha egy mágneslemezcsomag valamely felületén egy sáv megsérül, a tartaléksávok közül kell kijelölni egy újat, és a hibás sávot ki kell iktatni. Erre a célra is megfelelő program áll rendelkezésre.

Igen sokszor van szükség arra, hogy különböző mágnesszalagok vagy mágneslemezek tartalmát kilistázhassuk, állományokat másoljunk egy háttértárolóról egy másikra.

Hasonlóan gyakori igény a különböző könyvtárakban levő programok javítása. Mindezen feladatok elvégzésére is segédprogramok állnak rendelkezésre.

A segédprogramokat két nagy csoportba szokás sorolni. Ezek:

— a rendszerfüggetlen segédprogramok, amelyek akkor is működtethetők, ha nincs betöltve operációs rendszer;
— a rendszertől függő segédprogramok, amelyek az operációs rendszer felügyelete alatt futnak.

A rendszertől függő segédprogramok egyik csoportját képezik azok, amelyek a rendszer saját adatait és adathordozóit kezelik. A másik csoportba a felhasználók adatainak kezelését biztosító segédprogramok tartoznak.

Az elmondottakat két példával szemléltetjük. A rendszer saját adatai közé tartozik pl. a mágneslemezek elején található tartalomjegyzék. Ennek tartalma – a lemezen levő állományok neve és azok jellemzői – sokszor szükségesek az üzemeltető számára.

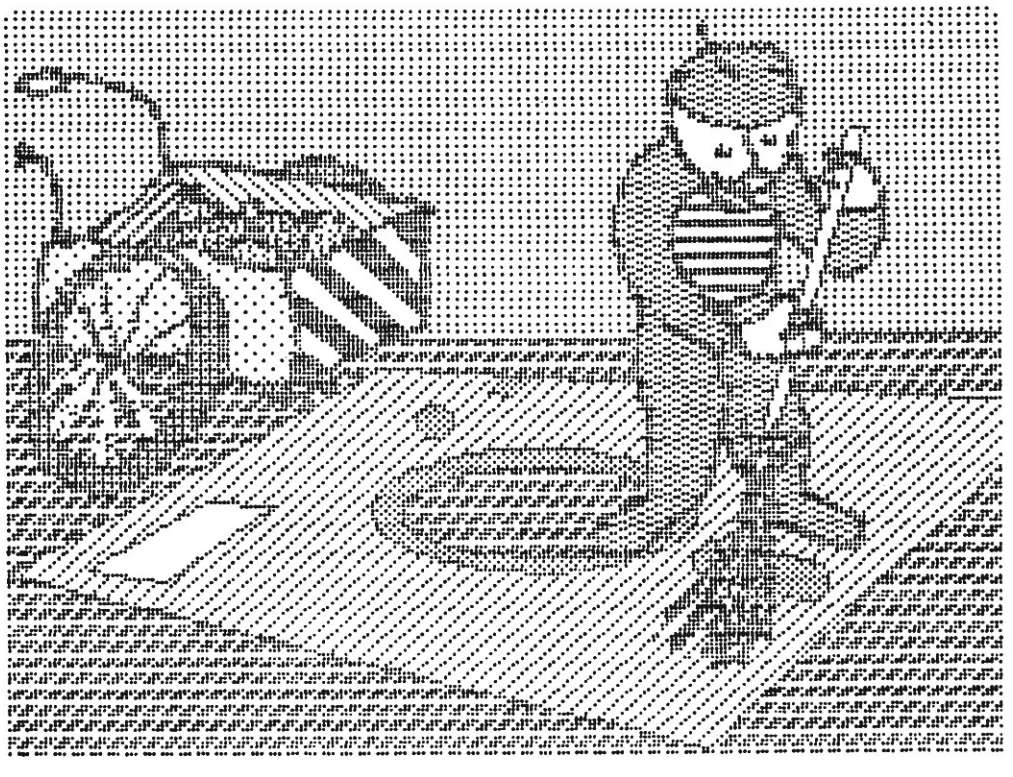
A tartalomjegyzék listázását biztosítja egy megfelelő segédprogram. A felhasználó számára a mágneslemezen levő könyvtárainak karbantartása fontos feladat, ha nem akarja, hogy azok beteljenek. Könyvtárak karbantartása (feleslegessé vált programok törlése, a könyvtárba „szétszórtan" levő programok egymás mögé tömörítése) egy másik segédprogrammal kényelmesen megvalósítható.

Ugyanezek a tevékenységek a személyi számítógépek esetében lényegesen egyszerűbbek. Ennek oka nem az, hogy ezek a gépek „okosabbak", hanem az, hogy egy egyfelhasználós rendszerben egyszerűbben oldhatók meg a feladatok. Úgy is fogalmazhatnánk, hogy a kezelni kívánt állományok közelebb vannak a felhasználóhoz, mint a nagyszámítógépes környezetben. Az 50. fejezetben megmutatjuk, hogy az előbbi két feladat milyen egyszerűen oldható meg személyi számítógéppel.

Vannak persze olyan problémák is, amelyek célszerűen segédprogrammal oldhatók meg, és a megoldás mind a nagy- mind a mikrogépeken egyformán történik. Tipikusan ilyen feladat egy állomány rekordjainak rendezése. Viszonylag gyakran előforduló probléma, hogy egy feldolgozás, vagy listázás előtt bizonyos adatokat, adatcsoportokat valamilyen szempont szerint rendezni kell. Ez lehet például ábécé szerinti rendezés, vagy (például sporteseményeknél) elért eredmény szerinti rendezés stb.

A rendezendő állományban azonos felépítésű adatcsoportok (rekordok) vannak. Ez azt jelenti, hogy a név, vagy az elért pontszám (eredmény) minden résztvevő rekordjában azonos helyen, azonos pozíción van. A rendezést végző programmal csupán azt kell közölni, hogy mit csináljon. Így meg kell mondani, hogy mi a neve a rendezendő állománynak, mi legyen a neve a rendezett állománynak. Közölni kell, hogy növekvő sorrendű rendezést akarunk-e (pl. az ábécé-rendezés ilyen), vagy csökkenő sorrendűt (ilyen lehet a verseny-eredménylista, ha a magasabb pontszám jobb helyezést jelent). Meg kell ezen kívül adni, hogy az az adat (név, pontszám), amelyre a rendezés vonatkozik melyik pozíción kezdődik, és milyen hosszan tart. A program ezek alapján (szükség esetén több ismérv kombinációját is figyelembe véve, pl. azonos pontszámon belül ábécé-sorrend) gyorsan elvégzi a rendezést.

Segéd-program

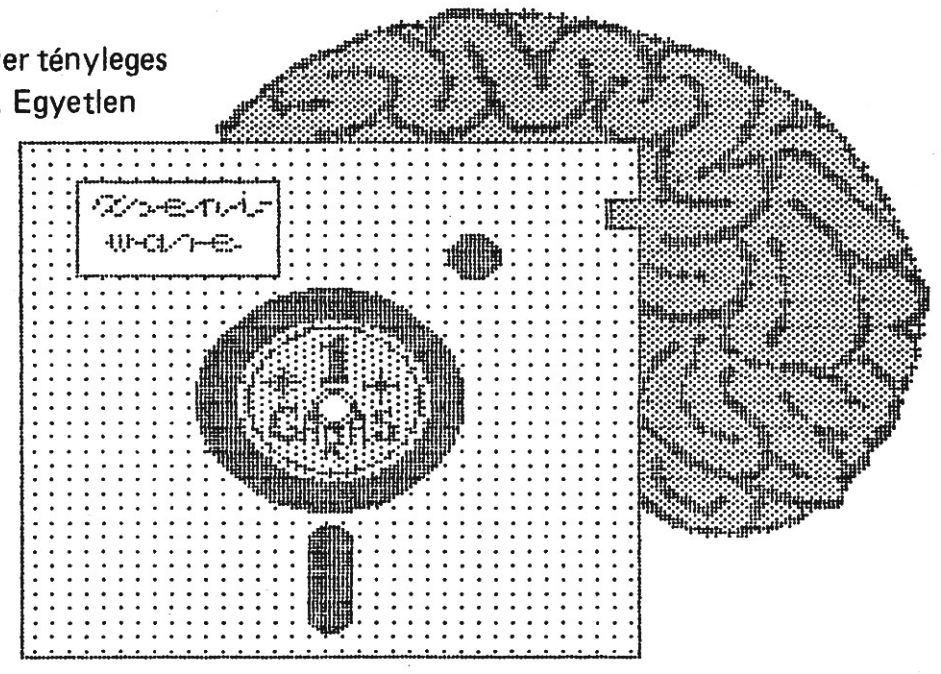


A sebesség azért fontos, mert egy roszszul megírt rendezőprogram akár tízszer lassúbb lehet egy jól megírt sebességéhez képest. Ez is arra figyelmeztet, hogy csak akkor magad, ha segédprogramod nincsen.

37.

A SZOFTVER ÉRTÉKE

Egy számítógép megvásárlásakor a vételárban nemcsak a hardver árát kell megfizetni, hanem annak az úgynevezett alapszoftvernek az értékét is, amely mindenképpen szükséges ahhoz, hogy a gépet egyáltalában használni lehessen. A régebbi berendezéseknél még elsősorban a hardverköltségek domináltak, de az utóbbi évtizedben ez az arány fokozatosan eltolódott a szoftver javára. Olyannyira, hogy ma már a gyártó cég által a géppel együtt szállított és annak árában benne levő látszólag „ingyen" szoftver, vagy az üzemeltetéshez ajánlottan, vagy kötelezően, de mindenképpen célszerűen megveendő szoftver értéke a teljes vételárnak a felét is kiteheti. Emellett a legtöbb cég a gép hatékonyabb kihasználását elősegítő programokat (pl. az adatok gyors rendezését, programok optimális fordítását elvégző programokat) is csak külön pénzért bocsátja a felhasználó rendelkezésére. Igen jól megélnek az alapszoftver és általánosan felhasználható, gyakran előforduló feladatokat megoldó programcsomagok készítésére szakosodott szoftverházak, melyek az egyszer kifejlesztett programot több felhasználónak is el tudják adni, így a fejlesztés költségei megoszlanak.



Ez azonban a szoftver tényleges árának csupán egy része. Egyetlen számítóközpont sem dolgozik kizárólag készen vett programokkal, sőt, teljes költségvetésüknek a feladatorientált alkalmazási programcsomagok vételi-, vagy bérleti díja nagyságrendben csak néhány százalékát köti le. Ugyanakkor viszont a számítóközpontokban dolgozó emberek munkabére a számítástechnikailag fejlett ipari államokban az összes költségnek mintegy felét teszi ki, ami körülbelül másfélszerese az üzemeltetett hardver amortizációjának. Ha figyelembe vesszük, hogy az alkalmazottak hanyadrésze foglalkozik üzemeltetéssel illetve új feladatok megoldására szolgáló rendszerek fejlesztésével, akkor jogos az a becslés, hogy a számítóközpontok összes kiadásuknak legalább 1/4-ét a „házi szoftver" előállítására fordítják. Hogy ez abszolút értékben mekkora lehet, azt az mutatja, hogy Magyarországon egy nagy számítóközpont évi költségvetése a beruházásokat leszámítva is tízmilliós nagyságrendben mozog.

Hátha a kedves olvasó az eddigiek alapján kedvet kapott ahhoz, hogy ne csak „műkedvelő" számítástechnikusként foglalkozzon a szakmával, hanem ezzel keresse meg kenyerét, — és természetesen a rávalót is. Ezért néhány hasznos adatot szeretnénk ismertetni, a számítástechnikával foglalkozók hazai kereseti lehetőségeiről és a munkakörökről, illetve a foglalkoztatottak számáról.

A következő táblázat megmutatja, miként változott hazánkban 1982 és 1986 között a számítástechnikai szakmában foglalkoztatottak száma a különféle jellegű munkahelyeken és hogyan nőtt ugyanezen idő alatt az egy főre eső havi bruttó (levonások előtti) átlagjövedelem a hivatalos statisztika szerint. Az átlagjövedelmbe természetesen beleszámítanak a 4—6 ezer forintot kereső adatrögzítők is és a 15—25 ezer forint hivatalos jövedelemmel rendelkező szuper rendszerprogramozók is. 1985-ben a számítástechnikai szakemberek 3/4-ét foglalkoztató vállalatok, költségvetési intézmények, szövetkezetek száma 3000 volt, míg a számítástechnikai tevékenységet végző kis szervezetek száma (VGM, PJT, GMK) 1700-ra növekedett. A szakmában dolgozók összlétszáma — figyelembe véve a részmunkaidős foglalkoztatást is — 1984-ben 25,5, 1985-ben 27, 1990-ben pedig várhatóan 31 ezer fő lesz.

A számítástechnika területén foglalkoztatottak számának és havi bruttó átlagjövedelmének alakulása Magyarországon, 1982 és 1986 között

Év / Munkahely	1982		1983		1984		1985		1986**	
	Létszám	Jövedelem	Létszám	Jövedelem	Létszám	Jövedelem	Létszám	Jövedelem	Létszám	Jövedelem
Vállalatok, költségvetési intézmények szövetkezetek	20 047	5 003	20 891	5 251	21 694	5 886	21 662	7 300	22 000	8 300
Vállalati Gazdasági Munkaközösségek (VGM)*	539	3 165	1 848	3 890	3 565	3 780	5 734	3 500	5 500	3 900
Kisvállalatok kisszövetkezetek	93	4 770	271	6 910	1 210	7 710	2 510	11 400	4 500	13 000
PJT, GMK	—	—	—	—	5 275	5 040	5 613	7 430	5 600	8 400

*Részmunkaidőben foglalkoztatottak

**Előzetes adatok, becslés alapján

Ezeknek a számoknak láttán óhatatlanul felvetődik a kérdés, vajon valóban szükség van évente annyi új programra? Egészen biztos, hogy nincs! Egy-egy program — bizonyos átalakítással — többféle feladatnak a megoldására is felhasználható. Mégis szinte minden feladatra új programot írnak! A szellemi energiának ez a tékozlása a legutóbbi években világszerte csökkent, ám a jelenséget teljesen megszüntetni korántsem sikerült. A többször — és fölöslegesen — megírt programokban megtestesülő munkának az értékét ma 100 milliárd dollárra becsülik, s ez az összeg folyvást nő. Hogy miért? Egyrészt azért, mert a programozókat elsősorban arra tanították meg, hogy új programokat készítsenek, és ezért a munkáért is fizetik meg őket. Így azután inkább programokat írnak, mintsem hogy megtanulják, miként használhatók a már meglevők.

Másrészt a számítástechnikát alkalmazók körében még ma is tartja magát az a régebben talán helytálló, de ma már idejétmúlt nézet, amely szerint mindenki úgy véli: az ő problémája annyira sajátos, hogy arra nem találni a szoftverpiacon kész megoldást. Nagy összeget takaríthatnának meg, ha a programokat eleve olyan általánosan írnák meg, hogy azokat mások is könnyen felhasználhassák saját feladataik megoldásában. S végül: ha valahol elkészül egy új program, arról a hasonló feladatokon dolgozó számítóközpontok vezetőinek, szervezőinek és programozóinak tüstént tudomást kellene szerezniük.

38.

KALÓZKODÁS A TENGERNYI SZOFTVERBEN
(A szoftver jogi és fizikai védelme)

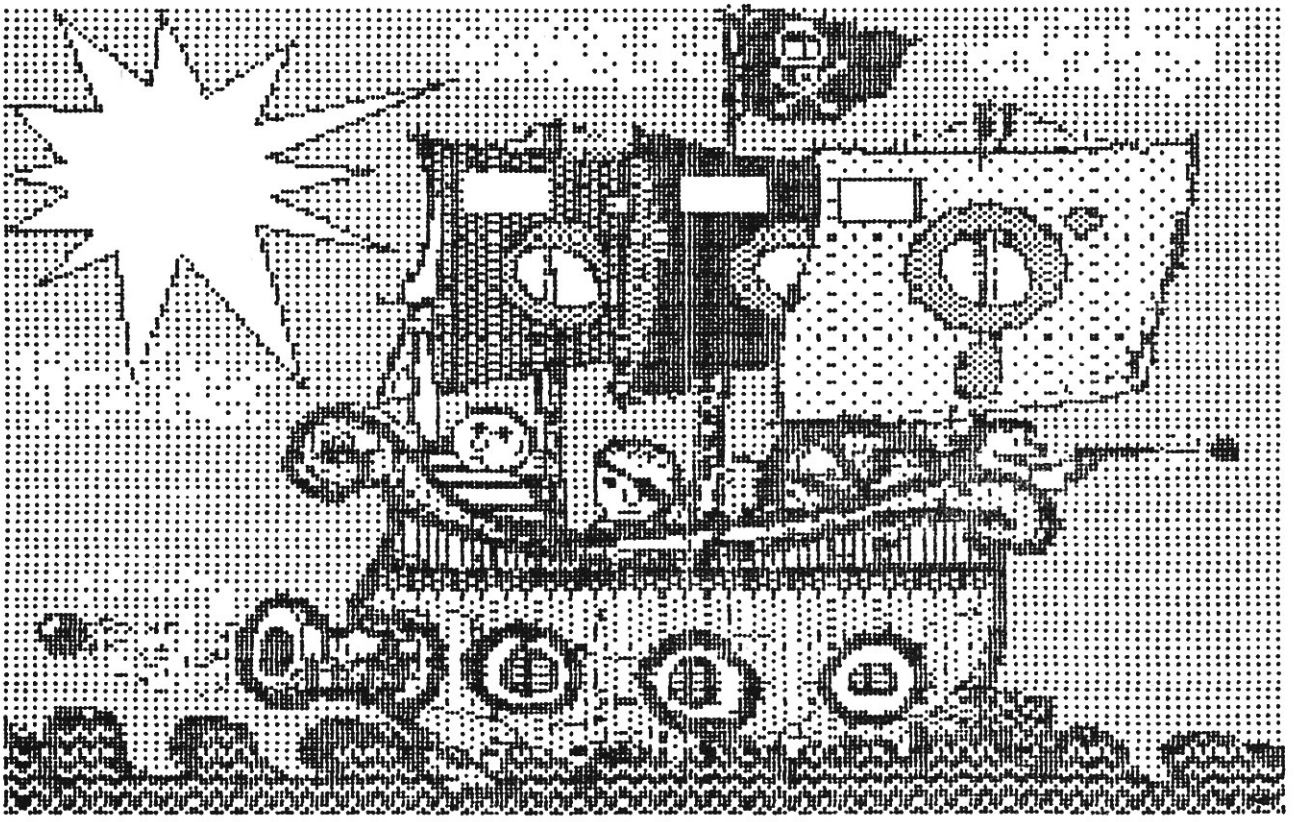
Átlag Jenő számítógépes programozó — csakúgy, mint könyvünk szorgalmas olvasója — tisztességes, becsületes állampolgár. Eszébe sem jut, hogy ellopjon egy számítástechnikai szakkönyvet, még kevésbé egy számítógépet. De könnyen meglehet, hogy minden lelkiismeretfurdalás nélkül átmásolja a saját kazettájára vagy mágneslemezére barátja pénzért megvett játékprogramját és otthon kiválóan szórakozik vele. Minden további nélkül használja munkahelyén azt a szövegszerkesztő vagy ügyviteli programot is, melyet egy másik vállalatnál dolgozó kollégája adott át neki cserébe az ő cégük által megvásárolt táblázatszerkesztő program egy másolatáért, — minek vegyük meg ugyanazt kétszer, ha egyszer is elég! — felkiáltással.

Sőt, ha egy számítógépes program nemcsak jogilag, hanem fizikailag is védett, azaz készítője megkísérelte lemásolhatatlanná tenni, akkor egyes programozók körében nemhogy bűn, de inkább dicsőség ezekről szabadon felhasználható változatokat készíteni. Pedig erkölcsileg és ma már jogilag is ugyanolyan bűnt követ el az, aki eltulajdonít egy számítógépet, mint aki jogtalanul, engedély nélkül használja a szerzői jog által védett programokat. Legfeljebb azon lehet vitatkozni, hogyha csupán saját szórakoztatására, saját munkájának megkönnyítésére teszi ezt, akkor bűne bocsánatos-e, és vajon csak azt kell-e szigorúan megbüntetni, aki a lopott programok másolatait pénzért árusítja.

A szellemi termékekkel való kalózkereskedelem, a legálisan vagy illegálisan megszerzett, védett eredeti anyagnak engedély nélküli sokszorosítása nemcsak a szoftvergyártásra és -forgalmazásra jellemző. Korábban könyvek, napjainkban inkább hang- és videószalagok százezrei forognak szerte a világban úgy, hogy ezek eladásának hasznából sem a szerzők, sem a jogosult forgalmazók egy fillért nem látnak. Ezt a pénzt a hang- és videókalózok teszik zsebre. A nagy üzlet magyarázata igen egyszerű. Egy ilyen jellegű szellemi termék mindig valamilyen — a termék szellemi értékéhez viszonyítva — nagyon olcsó hordozóközeghez kötődik, pl. papírhoz, kazettához, lemezhez. Az értékes termék átmásolása egy másik hordozóra általában szintén nem kerül sokba és ráadásul ez a művelet semmivel nem csökkenti az eredetinek az értékét, használhatóságát. Amikor egy üveg pálinkát áttöltenek egy másik palackba, vagy az egyikben vagy a másikban lesz az ital. Ha pedig felvizezték, akkor mindkettőben gyengébb lesz a minősége. De ha egy számítógépes programot át tudunk másolni az egyik lemezről egy másikra, akkor az mostmár két különböző lemezen lehet, méghozzá mindkettőn tökéletesen felhasználható állapotban. Ekkor pedig az egyik példány már eladható. Az eladónak a szellemi termék kifejlesztéséhez, a piacra való bevezetéséhez szükséges költségek helyett csak a hordozóanyagnak és a másolásnak az árát kell befektetnie az üzletbe. A nyolcvanas évek elején, amikor a mikroszámítógépek piaca virágzott és a hozzájuk tartozó programok legális úton való árusítása még ennél is nagyobb ütemben nőtt, a „hamisítás" nem jelentett komoly problémát. A piac telítettségével azonban egyre nagyobb gondot okozott a szoftverkészítőknek, hogy termékeiket nem tudták kellő példányszámban eladni. A vevők egy része — akarva vagy akaratlanul — az illegálisan lemásolt és forgalomba hozott másodpéldányok vala-

melyikét vette meg, melyet a forgalom növekedése érdekében gyakran az eredetinél lényegesen olcsóbb áron – de még így is tetemes haszonnal – hoztak forgalomba a hamisítók.

A bűnözés, valamint annak megelőzése, megakadályozása és jogi úton való megbüntetése mindig versenyfutást jelent. A szoftverkalózkodás területén is megérett a helyzet arra, hogy egyrészt technikai eszközökkel lehetetlenné vagy legalábbis olyan nehézzé, költségessé tegyék a tolvajok dolgát, hogy az üzlet az esetleges leleplezés várható jogi következményeit is figyelembe véve ne legyen már kifizetődő.*



A jogi védelem érdekében a legtöbb fejlett országban, így Magyarországon is törvényt hoztak, amely szerint a szerzői jogot kiterjesztették a számítógépes programokra is. Ennek értelmében, ha egy önálló szellemi munka eredményeként létrejött programot vagy programrendszert az előírásoknak megfelelő módon bejelentenek a Szerzői Jogvédő Hivatalnál, akkor az lényegében hasonló jellegű védelemben részesül, mint egy irodalmi vagy zenei alkotás. Tilos és büntetendő cselekedet tehát az így védett szoftvert vagy ennek másolati példányait a szerző vagy a hivatalos forgalomba hozó engedélye nélkül árusítani, bármilyen célra fölhasználni, sőt, van olyan ország, ahol már az is büntetendő, ha valakinek ilyen példány a birtokában van.

Természetesen minden törvény csak annyit ér, amennyire azt be lehet tartani. Azokban az országokban, ahol a szoftverjogokat szigorúan ellenőrzik, a nagy szoftverkereskedő cégek nem árusítanak illegális úton előállított másolatokat ugyanúgy, ahogy így beszerzett videó- és hangszalagokat sem forgalmaznak. Ez azonban nem zárja ki azt, hogy az egyes felhasználók a saját birtokukban levő programokat ne másolják le, és egymás között ne csereberéljék.

* Egy másféle módszer is lehetséges, melyet már néhány élvonalbeli szoftverház is alkalmaz: nagyon olcsón adja a szoftvert és a hozzátartozó jól használható kézikönyvet, és nem védi másolás ellen. A tapasztalat szerint ekkor jóval többen veszik meg az eredetit, s így is rentábilis az eredeti forgalmazó üzlete. (A lektor megjegyzése.)

Védtelenek a szoftvergyártók az olyan országokban, amelyekben a szerzői jogok ellenőrzése gyenge, és a jogtalan árusítás megbüntetése, különösen, ha a díjak külföldit illetnek meg, olyan nehézkes és hosszadalmas, hogy a tulajdonosok inkább veszni hagyják jogdíjaikat.

Éppen ezért az igazi védelem olyan, ami a programot engedély nélkül másolhatatlanná teszi, vagy az így másolt programot nem lehet használni. Az első módszer nagy hátránya, hogy ekkor a jogos felhasználók sem készíthetnek másolatot, holott ez biztonsági okokból nélkülözhetetlen. Történhet a program eredeti adathordozójával olyasmi, amitől az használhatatlanná válik. Sovány vigasz akkor, hogy a gyártó cég a meghibásodott eredeti bevonásával néhány napon vagy héten belül küld egy új, hibátlan példányt.

A fizikai védelem egyik formája az, hogy a hiteles lemez egy meghatározott helyére a szoftvergyártó valamilyen hibát helyez el. Ez lehet egy tényleges fizikai hiba, amelyet például lézerrel égetnek be, de lehet olyan is, hogy a lemez mágnesességének elrendezését változtatják meg úgy, hogy azt a gépek szokványos olvasóprogramjai nem tudják értelmezni. A védett programot úgy írták meg, hogy az a feldolgozás közben olvasni akar erről a helyről. Ha ott nem észlel rendellenességet, akkor nem hajlandó tovább dolgozni, míg hiba esetén azt elhanyagolja és folytatja működését. Ahhoz, hogy a lemásolt programot engedély nélkül használni lehessen, egy ugyanilyen hibát kell elhelyezni a másolaton is, vagy pedig ki kell iktatni a programból azt a részt, amelyik a lemezhiba ellenőrzését végzi el. Mindkettő igen nagy technikai felkészültséget és szoftverismereteket igényel. A védelem feltörése többe kerülne, mint egy szoftverpéldány megvétele. Csak akkor lenne kifizetődő, ha a másolatokat sok helyre lehet terjeszteni. Ezért ez a módszer az egyedi eltulajdonítás ellen a gyakorlatban biztos védelmet nyújt. Az ebből üzletet csinálókra — mivel komoly hasznot csak nagy példányszám esetén érhetnek el, így tevékenységüket szakmai körökben valamilyen módon hirdetni kell — pedig előbb-utóbb lecsap a törvény szigora. Angliában például két évi börtön is kiszabható a szoftverhamisítókra, az illegális szoftverrel kereskedőkre.

Ennek a módszernek a vevő szempontjából lényegesen javított változata az, amikor a védett program a szándékos hibát tartalmazó rész átugrásával akárhányszor átmásolható. A másolatról működő program csak induláskor kéri az eredeti adathordozót. Ha ott észleli a hibát, tovább dolgozik, különben megáll. Ilyen módon az eredeti lemez sokkal kisebb igénybevételnek van kitéve, tovább használható, egy másolat meghibásodása nem okoz gondot. Ugyanakkor, aki nem vásárolta meg az eredeti adathordozón a programot, semmit sem tud kezdeni a birtokában levő másolattal.

Hasonló elven szolgáltat védelmet az a módszer, amikor a szoftverrel (programlemezzel) együtt egy kis dobozba elhelyezett, speciális funkciókat tartalmazó integrált áramkört is adnak. Ezt a számítógép szokványos kimeneteinek egyikéhez kell csatlakoztatni. A program ennek a meglétét ellenőrzi.

Ugyancsak jó védelmet nyújt az az eljárás, amikor a program indulásakor egy véletlenül kiválasztott többjegyű számot jelenít meg néhány másodpercig a képernyőn, de olyan össze-vissza kevert állapotban, hogy azt csak egy speciális, a programmal együtt árusított prizmán keresztül lehet leolvasni. Akinek a prizma a birtokában van, az beüti ezt a számkombinációt, mire a program elindul. Akinek nincs, az e rövid idő alatt, amíg a program még vár a válaszra, nem tudja ezt megtenni. Hiába próbálja újra, addigi tapasztalatait nem tudja semmire sem használni, hiszen akkor már egy egészen más számot kell felismernie ismét nagyon rövid idő alatt.

Természetesen ezek és más védekezési módszerek sem nyújtanak száz százalékos biztonságot a jogosulatlan felhasználás ellen. Nincs az a zár, amelyet kellő fortéllyal, munkával, szakértelemmel ne lehetne feltörni. Legföljebb az a kérdés, hogy az ebből származó

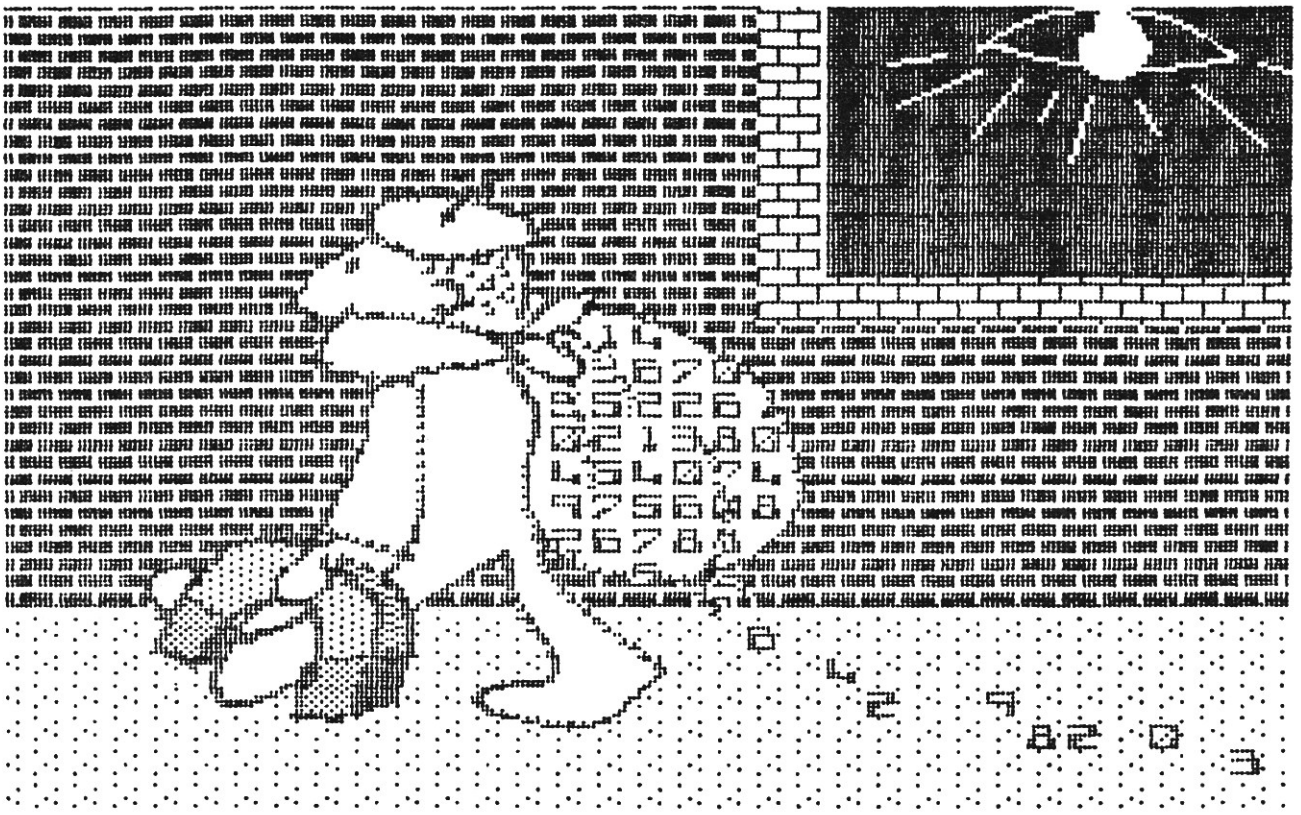
haszon megéri-e a vele járó fáradtságot és kockázatot. Minél több programra van szükség, annál inkább nagyüzemivé, olcsóbbá válik annak előállítása. Minél több helyen lehet felhasználni ugyanazt a szoftvert, minél több példányban lehet eladni, annál alacsonyabb lesz az egyes példányok önköltsége, ára. Minél olcsóbban lehet a programokhoz legálisan hozzájutni, annál kevésbé lesz kifizetődő az üzlet a hamisítóknak. Bár sokkal nehezebb, száz dollárost sokan próbálnak hamisítani, villamosjegyet valószínűleg senki sem. Talán a programokkal való kalózkodásnak is a szoftver árcsökkenése fog egyszer és mindenkorra véget vetni, amint erre már vannak jelek.

39.

VIGYÁZAT, ADATTOLVAJ!

(Számítógépes rendszerek biztonsága)

Pereg a film. Egy fiatal diák személyi számítógépe az adathálózat telefonvonalain keresztül véletlenül bekapcsolódott a NATO rakétaelhárító rendszerébe, és elindított egy ellenséges támadást szimuláló programot. A számítógép érzékeli az „ellenség" közeledését, és riadóztatja a rakétaelhárító központot. Eddig még mindenki csak szokványos hadgyakorlatnak tekinti az eseményeket. De a számítógép-rendszerbe véletlenül beavatkozó fiú nem tudja, miként kell befejezni a játékot, hogyan lehet lefújni a hadgyakorlatot.



A gép által vezérelt automatika megállíthatatlanul működésbe lép. Hiába tudják az emberek, hogy a valóságban nincsen semmiféle támadás, a számítógép a benne tárolt utasítások szerint átvette az irányítást. A központi rendszertől tűzparancsot kapnak a megtorló nukleáris rakétákat vezérlő számítógépek, kinyílnak a tárolósilók tetői. Már majdnem útnak indulnak a sok száz megatonna robbanóanyagot hordozó rakéták, amikor végül sikerül előkeríteni a rendszer főprogramozóját, aki megfelelő utasításokkal az utolsó percben leállítja a nukleáris visszacsapást megindító programot. Goethe bűvészinasának modern változata. Sci-fi, — de egyben jelenleg is elképzelhető valóság, melyet a néhány éve készült, külföldön nagy sikert aratott „Harci játékok" című film dolgoz fel lélegzetelállítóan izgalmas formában.

Rémálom, de ugyanakkor reális lehetőség is. Többször tudomást szerezhettünk arról, hogy a NATO számítógépei tévesen helyezték magasfokú készültségbe a rakétaelhárító

rendszert. Mi történne, ha egyszer nem, vagy csak későn vennék észre, hogy vaklárma volt a riadó? – kérdik a számítógéppel irányított rendszerek ellenzői és a békére vágyó emberek. Mennyire biztos ez a rendszer, bizonygatják a hívei – hisz eddig még mindig észrevette a saját hibáit, és ennek megfelelően intézkedett. Valóban, eddig ez így történt, de vajon így lesz-e mindig a jövőben is?

Az Egyesült Államokban naponta sok milliárd dollárt utalnak át a bankok számítógépei egymás között, és nálunk is több millió forint cserél gazdát a Magyar Nemzeti Banknál vezetett vállalati folyószámlákon pusztán utólagos emberi ellenőrzéssel. Az 1980–81-es iráni túszdráma váltságdíját minden ellenőrzést beleszámítva néhány perc alatt írták jóvá a különböző földrészeken levő számítógépek. Mi történne, ha egyszer a betörők nem a bank készpénzét és aranyát vinnék el a pénzcélszobából, hanem a számítógép tárolójában adatként őrzött pénzösszegekből utaltatnának át programmal hatalmas összegeket saját számlájukra, majd a készpénz felvétele után eltűnnének. Bár ilyen lopás lehetősége fennáll, és filmen már láthattunk is hasonló történetet, az életben való megvalósításához legalább akkora technikai felkészültség szükséges, mint például a több millió dolláros nizzai bankrabláshoz, amikor a föld alatt alagutat fúrtak a pénzcélszobához. A különbség legföljebb annyi, hogy itt nem a fizikai, hanem a logikai utat kell megtalálni és nem a fizikai kincstárolóhoz, hanem a számítógép adattárolójához és az átutalásokat irányító programrendszerhez kell eljutni.

Néhány évvel ezelőtt az egyik New York-i kórházban egy ismeretlen a számítógépeket összekötő adathálózaton keresztül kapcsolatba lépett a rákos betegek besugárzást irányító és a besugárzott dózisok nagyságát tároló számítógéppel, és tönkretette az ott levő programok és adatok jelentős részét. Ezt többször megismételte, míg végül a telefonvonalak és a rajtuk keresztül érkező utasítások együttes figyelése eredményeként az FBI azonosította és letartóztatta a tettest, – egy 21 éves fiatalembert, aki Wisconsinból, több mint 1000 km távolságból „játszadozott" egy kicsit.

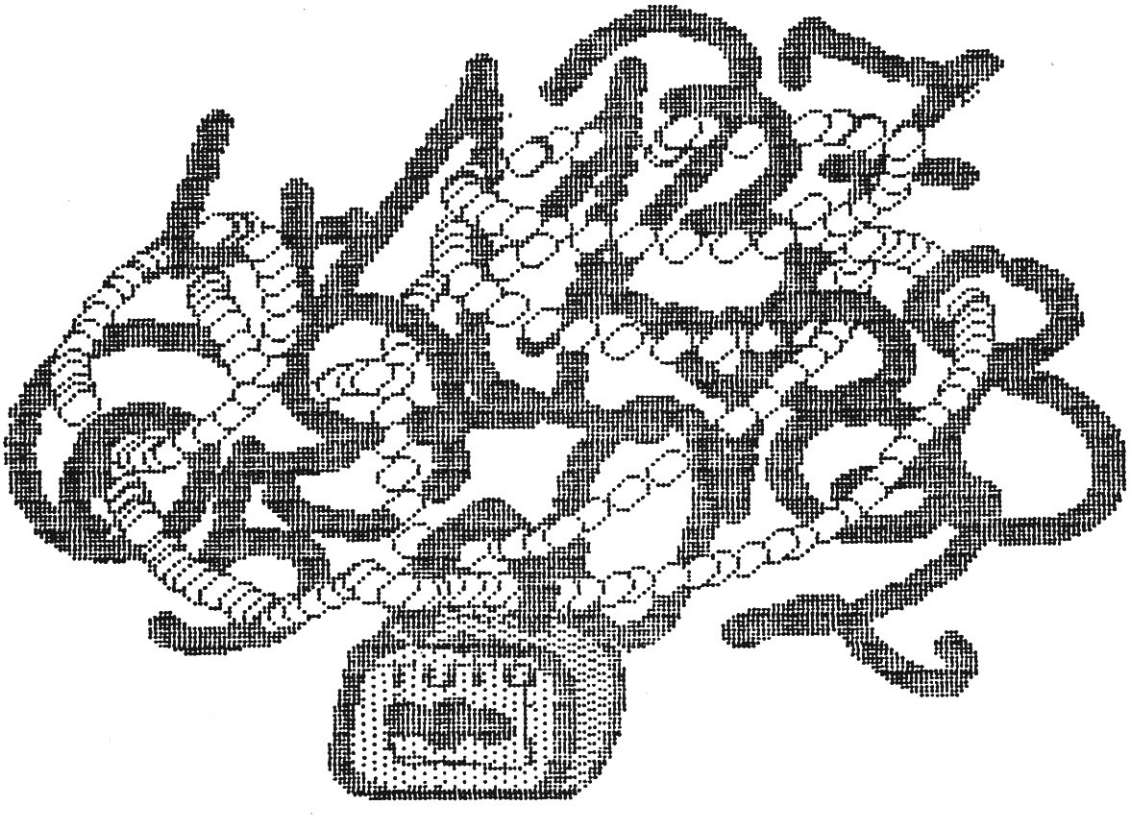
Korunk társadalmának információigénye olyan nagy, hogy ezt ma már csak számítógépekkel lehet kielégíteni. Az adatokat azonban nem elég csupán összegyűjteni. Ügyelni kell arra is, hogy azokhoz csak az arra jogosultak férhessenek hozzá. Emellett arra is kell vigyázni, hogy a létfontosságú adatok minden körülmények között (bármilyen erőszakos cselekmény vagy természeti csapás bekövetkezte után is) rendelkezésre álljanak. Ahogy nőtt a számítógépes adatbankok száma, és a bennük tárolt adatok és az ezekből nyerhető információ mennyisége, úgy növekedett az igény arra, hogy ezeknek biztonságát és titkosságát a lehető legnagyobb mértékben szavatolni lehessen.

A napi sajtó, a rádió, a tv egyre gyakrabban ad hírt a számítógépek segítségével elkövetett különböző bűncselekményekről. Alig, vagy csak igen ritkán hallunk viszont azokról a módszerekről, melyek megakadályozzák, vagy legalábbis igen nagy mértékben megnehezítik elkövetésüket és arról, hogyha mégis bekövetkeznének, miként lehet az így okozott kárt minimálisra csökkenteni. A következő fejezetben ezeket kívánjuk röviden ismertetni. Az adatoknak titkosított formába való átírásával a 63. fejezetben fogunk foglalkozni.

40.

ADATVÉDELEM, ADATBIZTONSÁG

Tökéletes biztonság a számítógépes rendszereknél sem létezik. Krőzus vagyona sem lenne elég ennek megvalósításához. Minden olyan tevékenység, mely egy számítógépes rendszer folyamatos, biztonságos és zavartalan működésének valószínűségét növeli, erőforrás- és ezáltal pénzigényes. A gépek és berendezések, a hardver biztonságos üzemelését nagymértékben elősegíti, ha a létfontosságú berendezések mellett mindig ott áll üzemkészen egy tartalék is. Igen ám, de még egy fontos pótberendezést pótló pótegység is meghibásodhat és akkor ennek helyettesítéséről is azonnal gondoskodnunk kellene. Ezen meggondolás alapján tökéletes biztonsághoz minden berendezést kétszeresen, háromszorosan, . . . sokszorosan kellene megvásárolnunk, és állandóan üzemkész állapotban tartanunk. Ez nyilvánvalóan értelmetlen és lehetetlen lenne.



A vezérlő és a feldolgozó programokba beépített ellenőrző-tevékenységek és az illetéktelen hozzáférési kísérleteket kijelző eljárások elkészítése magasan képzett szakemberek drága munkáját igényli. Emellett üzemközbeni folyamatos végrehajtásuk jelentős időbe kerül, és komoly gépi kapacitást köt le, melyeknek költségei szintén nagy összeget tesznek ki. Jól telepített rendszernél ez annál nagyobb, minél tökéletesebb biztonságot nyújt ez az ún. szoftvervédelem.

Nyilvánvaló, hogy a biztonsági berendezések és eljárások költségének valahol határt kell szabni. De hol?

Ha azt akarjuk, hogy a lakásunkba illetéktelen személy ne léphessen be, kulcsra zárjuk az ajtót. Ha értékesebb holmijaink vannak, akkor több biztonsági zárat és rácsot szereltetünk el. Sőt, ha még nagyobb biztonságra törekszünk, akkor ezeken kívül még olyan riasztót is beépíttetünk, amely illetéktelen behatolás esetén fényképet készít a betörőről, és egyben riasztócsengőt szólaltat meg a rendőrségen. Magánlakásban általában már nem kifizetődő az ilyen drága rendszer, de bankokba, múzeumokba már érdemes beszerezni, még a lopás megtörténte után is, – föltéve, hogy üzembe is tudjuk helyezni.

A biztonságosan üzemelő számítástechnikai rendszerek megvalósításánál arra kell törekedni, hogy a káros esemény bekövetkezésének valószínűségét igen alacsonyra szorítsuk le, és ha mégis bekövetkezne a baj, akkor az okozott kár mértéke minél kisebb legyen. A bekövetkezés valószínűségének és a kár nagyságának a szorzata kell arányban álljon a biztonsági intézkedések és berendezések beszerzési és üzemeltetési költségeivel.

rendszerek biztonsága

Amikor a szakemberek számítógépes információ- vagy adatfeldolgozó *rendszerek biztonságáról* beszélnek, akkor a számítógépek mellett a rendszerhez tartozónak tekintik a gépet kiszolgáló személyzetet, azokat az adatokat és programokat, amelyekkel a feldolgozás történik, valamint korunk modern számítógépes hálózatainál azokat az adatátviteli vonalakat, vezérlőberendezéseket és adatvégállomásokat (terminálokat), amelyeken keresztül nagy, esetleg több száz kilométer távolságból történik a feldolgozás. Manapság már nem olyan ritka az a megoldás sem, hogy különböző városokban, sőt országokban működő számítógépes hálózatokon dolgoznak egyidejűleg ugyanazokon a feladatokon jól szervezett munkamegosztással. A gépek információkat cserélnek, kisegítik vagy helyettesítik egymást, ahogy az igények változnak a különféle helyeken.

Amíg a számítógépektől csak abból a helyiségből lehetett adatokat lekérdezni, amelyikben a gép állt, és a kapott információk is csak itt jelentek meg, addig a biztonság követelményeit tökéletesen kielégítette, hogy az illetékteleneket nem engedték be a gépterembe, és közvetlenül a gép mellett csak megbízható személyek dolgozhattak. Napjainkban azonban a számítógépes rendszerek biztonságát és titkosságát sokkal tágabban kell értelmeznünk. A korszerű biztonsági intézkedéseknek ki kell terjedniük

- magára a számítógépre és a hozzá közvetlenül kapcsolódó adattároló és megjelenítő berendezésekre,
- a számítógépet vezérlő és a feldolgozást, illetve információszolgáltatást elvégző programokra,
- a számítógépet vezérlő üzemeltető személyzetre,
- az információkat a számítógéptől távolabbi helyekre eljuttató adatátviteli hálózatra.

Az adatfeldolgozó és információrendszerek biztonságáról csak akkor beszélhetünk, ha biztosítjuk a rendszer összetevőinek külön-külön és egymással kölcsönhatásban is a biztonságát. Minél nagyobb teljesítményű egy rendszer, annál nehezebb feladata van annak a karmesternek, aki a biztonságos működés érdekében összehangolja a rendszer egyes részeinek működését. A számítástechnikai rendszerek biztonságánál is ugyanaz a szabály érvényes, mint egy láncnál. Mindig a leggyengébb láncszem határozza meg a rendszer biztonságát. Hiába vásárolunk vagy készítünk drága pénzért ellenőrző programokat, melyek tökéletes biztonságot nyújtanak arra, hogy illetéktelenek ne férhessenek a gépen keresztül az adatokhoz, ha a számítógép által valamilyen okból selejtként kinyomtatott eredményeket (pl. hibásan volt a formanyomtatvány behelyezve) takarékosságra való hivatkozással átadják a MÉH-nek.

A modern számítástechnikai rendszereket ma már úgy készítik a gyártó cégek, hogy a fejezet elején levő példában szereplő kulcsok, észlelő- és riasztó-berendezések mind hozzá-

tartoznak a rendszerhez, hardver- vagy szoftverszinten vagy ezek valamilyen kombinációjaként. Ezeket, kiegészítve a felhasználó esetleges saját külön védelmi előírásaival, együttesen *fizikai és logikai védelemnek* nevezzük, melyek biztosítják, hogy a központi számítógép mindig tudja, kinek a munkája van folyamatban, ki és honnan kapcsolódott a gépre, milyen témán kíván dolgozni, milyen programokhoz, illetve adatokhoz kíván hozzáférni, esetleg módosítást végezni, milyen gépi erőforrásokat (központi vezérlő egység idő, memória, mágneslemez, szalag stb.) kíván felhasználni. Ezeket összehasonlítja a memóriában tárolt táblázatban levő adatokkal, mely minden felhasználóra tartalmazza, hogy milyen erőforrások használatára jogosult, és csak a számára engedélyezettek végrehajtását teszi lehetővé. Azokban az országokban, melyekben a számítástechnikát elterjedten alkalmazzák, így hazánkban is külön törvény szabályozza, kinek miről van joga információkat szerezni a különféle központi állami adatbankokból, és hogy milyen módon védekezhetnek az állampolgárok az ellen, hogy róluk illetéktelenek beleegyezésük nélkül adatokat gyűjthessenek össze.

fizikai-, logikai védelem

Az *illetékesség* vizsgálata akkor a legfontosabb, ha a rendszerhez való hozzáférés távolról, terminálon keresztül történik. Az ellenőrző program első teendője az, hogy azonosítsa a terminálon bejelentkező személyt. Ez leggyakrabban jelszóval és/vagy azonosító kártyával történik.

illetékesség, jogosultság-ellenőrzés, titkosság

A *jelszó* egy csupán a jogosult felhasználók által ismert betű- és számkombináció. Ezt a felhasználó a terminál billentyűzetén keresztül viszi be a rendszerbe. Azonosítás és *jogosultságellenőrzés* után a rendszer vagy engedélyezi, vagy megakadályozza a hozzáférést. A *titkosság* megőrzése érdekében — ellentétben a terminálon keresztül bevitt egyéb szöveggel — az így megadott jelszó soha sem jelenik meg sem a terminál képernyőjén, sem a vele esetleg összekapcsolt nyomtató berendezésen.

Más megoldásnál az azonosító kártyát a felhasználó behelyezi a terminál leolvasó nyílásába. A kártyán levő mágnescsíkról a gép leolvassa a felhasználó azonosítókódját, és ennek alapján ellenőrzi a hozzáférés jogosultságát.

Terminálóknál a védelem kiegészíthető még egy fizikai zárral is, mely pl. mechanikusan megakadályozza a készülék feszültség alá helyezését.

A biztonság növelése érdekében gyakran alkalmazzák a jelszó és az azonosító kártya kombinációját is. Ezt használják például a nyugaton már széles körben elterjedt, éjjel-nappal működő automata bankok. A számlatulajdonos bedugja az azonosító hitelkártyáját a leolvasóba, és beüti a billentyűzeten a jelszavát valamint a kért összeget. Ezeket és a kártyáról leolvasott csekkszámlaszámot az automata pénzkiadó továbbítja a központi nyilvántartóba. Ott a gép ellenőrzi, hogy a jelszó és a számlaszám összetartoznak-e, és hogy van-e fedezet a számlán. Amennyiben igen, úgy a pénzt (egy bizonyos maximalizált összegig) azonnal kiadja, akár éjjel van, akár vasárnap.

Nagy rendszereknél a jelszó rendszerint többlépcsős. Általában minél több fokozatát ismeri valaki, az adatállomány annál nagyobb részéhez férhet hozzá és annál többféle műveletet végezhet el az adatokon. Így például a felhasználó az egyik jelszóval jelentkezik be a terminálról. Egy másik jelszóval olvashat adatokat egy bizonyos állományból, és csak egy harmadik jelszó beadásával nyílik lehetősége arra, hogy ebbe újakat vihessen fel, vagy meglevőket töröljön. Komplex lekérdezési rendszereknél szükségessé válhat a terminálok azonosítása is. Ezt viszonylag könnyű megoldani akkor, ha a terminál fizikai elhelyezése behatárolja a felhasználás módját. Például a vállalat pénzügyi adatait csakis a Pénzügyi Osztályon elhelyezett terminálról lehet módosítani úgy, hogy a bevitt adatokról egyidejűleg írásos bizonylat is készüljön a géphez kapcsolt nyomtatón.

hozzáférési jog

Hogy a titkosság, a *hozzáférési jog* szabályozása milyen bonyolult problémákat vethet fel, annak illusztrálására tekintsük a következő példát. Egy személyi nyilvántartás során a felhasználókat az alábbi kategóriákba kívánjuk besorolni:

a) semmilyen adathoz nem férhet hozzá;
b) minden adathoz korlátozás nélkül hozzáférhet, és bármiféle műveletet (beleértve a változtatást is) végezhet rajtuk;
c) minden adatot elolvashat;
d) minden adatot elolvashat a „fizetés" és a „bizalmas információk" mezők kivételével;
e) csak a saját adatait olvashatja el;
f) mindenféle műveletet végezhet a saját adatain a „fizetés" és a „bizalmas információk" mezők módosításának kivételével;
g) csak a „név", „azonosítószám" és „fizetés" mezőkhöz férhet hozzá, és az utóbbit egy adott terminálról módosíthatja, de kizárólag munkaidőben;
h) statisztikai műveleteket végezhet a „fizetés" mezőn más mezők (pl. „nem", „osztály", „végzettség") értéke lapján, de az egyes fizetéseket nem olvashatja és semmit nem módosíthat;
i) az osztályon, melynek ő a vezetője, az összes dolgozó minden adatához hozzáférhet.

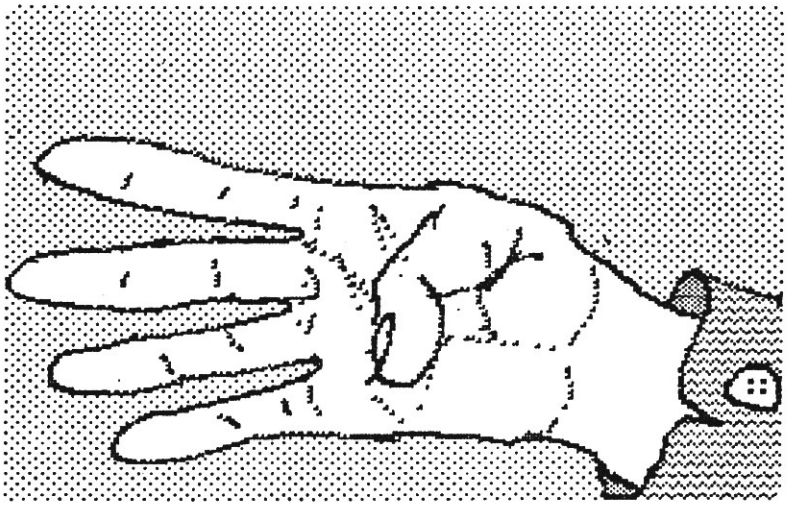
Természetesen ez a lista nem teljes, csupán a sokféle variációs lehetőségről ad némi ízelítőt. A hozzáférési jogok ennek alapján való szabályozása a mai korszerű rendszerekben megvalósítható, csupán az a kérdés, hogy ennek költségei (melybe nemcsak a kérdés feltételekor történő ellenőrzésre felhasznált gépidőt kell beleszámítanunk, hanem az ellenőrzéshez szükséges adattöbbletet és a bonyolultabb, ennek következtében általában kevésbé hatékony rendszert is) arányban állnak-e a titkosság ilyen részletes szabályozásával.

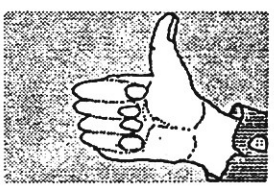
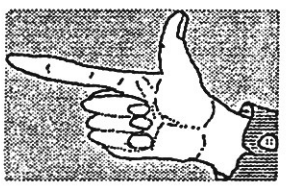
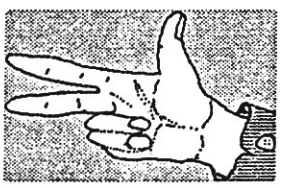
Azt, hogy kit milyen hozzáférési jog illet meg, magasabb szinten, rendszerint az intézmény legfelső vezetésének szintjén kell eldönteni. A jelszavakat ellenőrző tényleges hozzáférési lehetőségeket lehetővé tevő programokat aztán az adatbank működéséért felelős vezető szervező-programozó építi a szoftver rendszerbe.

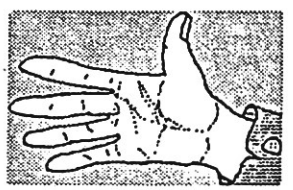
A rendszer biztonsága nemcsak azt követeli meg, hogy jól működő „reteszrendszert" alakítsunk ki, hanem azt is, hogy észleljünk minden olyan akciót, ami ennek megsértésére vagy megkerülésére irányul. A gyakorlatban használt megoldásokból összegyűjtöttünk egy párat:

a) Hamis jelszóval vagy kártyával történő próbálkozás esetén azonnal vagy másodszorra riasztó jelzést ad a gép a rendészetnek.
b) a második vagy harmadik egymás utáni hamis jelszóval történő próbálkozás után „lezár a terminál" úgy, hogy azt csak az arra illetékes személy tudja újra bekapcsolni.
c) A rendszer kérdésekkel foglalja el az illegálisan próbálkozót – vigyázva arra, hogy a bizalmas információkhoz ne juthasson – és közben értesíti a rendészetet vagy a rendőrséget, akik a helyszínen tetten is érhetik az illetőt.

DOLGOZ(Z)UNK
A SZÁMÍTÓGÉPPEL !





41.

AKINEK AZ ISTEN GÉPET ADOTT . . .

(Miért nehéz programot írni?)

Az ismert mondás szerint akinek az isten hivatalt adott, annak észt is adott hozzá. E mondás modern változata — amelyet nem mi találtunk ki — így hangzik: „Akinek az isten (vagy bárki más) számítógépet adott, annak programozási képességet is adott hozzá".

Valóban ilyen egyszerű volna-e a világ, hogy a programozáshoz nem kell más csak egy számítógép, meg egy leírás arról a nyelvről, amelyet használni akarunk? Általános iskolás gyermekeink magabiztossága, ahogyan a géppel bánnak, igenlő választ sugall. De vajon ez nem a hályogkovács biztonsága? A hozzá nem értő emberé, aki nem tudja, mit ronthat azzal, ha nem javít?

Megírni egy apró programot és megtervezni, elkészíteni egy nagy rendszert — ezek nem azonos súlycsoportba tartozó dolgok, mint ahogy nem azonos feladat 100 métert futni, vagy ugyanezt időre megtenni. Minderről az 52. fejezetben olvashatunk részletesen. Ami most és itt érdekel bennünket, az a következő probléma: Miért nehéz jól megírni egy programot, miért nehéz egy megírt program jóságáról biztosan meggyőződni?

Számítógépen futtatandó programok listájának tanulmányozása során sok olvasónk tapasztalhatta a „saját bőrén" azt a kettősséget, amely az általuk vizuálisan látható program és a gép által végrehajtott program között van.

Az ember számára egy program mindig az utasítások leírt sorrendjében látható. Nevezzük ezt az utasítássorrendet a továbbiakban statikus programnak. Ugyanennek a programnak az utasításai a végrehajtás során (a programban levő vezérlésátadások miatt) nem ilyen sorrendben követik egymást. Legyen ez a sorrendiség a továbbiakban a dinamikus program.

A probléma tehát az, hogy a programozónak a statikus programból kell a dinamikusra következtetnie. Ezt alapvetően két dolog nehezítheti. Az egyik a program mérete, a másik a program belső szerkezete.

Nyilvánvaló, hogy minél hosszabb egy program, annál nehezebb az imént említett következtetés (a program megértése). De az már nem mindenki számára nyilvánvaló, hogy itt nem egyenes arányról van szó. Gondolkodásunk akarva akaratlan a teljes indukciót követi: ha egy 50 utasításos program megértése 5 percet igényel, akkor 500 vagy 5000 soros programé tízszer, százszor ennyi időt. De ez nem igaz! Gondolkodásunknak ugyanis határai vannak. A programozó felkészültségétől függően mindig létezik egy utasításszám-határ, amely felett a megértés már nem pusztán mennyiségi kérdés. Ezen a határon túl ugrásszerűen megnő a hibázás valószínűsége, illetve a meglevő hiba fel nem fedezésének valószínűsége.

A szakirodalom 50 és 100 közé teszi azt az utasításmennyiséget, amelyet még jól át lehet tekinteni.

Ebből következik az egyik fontos programozási elv: azt a feladatot, amelynek mérete nagyobb lenne 50—100 utasításnál, részekre kell bontani, és a részeket önálló modulokként kell elkészíteni, majd a végén ezeket összefűzni. (Minden valamire való progamozási nyelvnél megvannak az ehhez szükséges eszközök.)

A másik probléma ami nehezítheti a statikus program alapján a dinamikus program megértését, a program belső szerkezete. Minél több vezérlésátadás van egy programban (főleg "GO TO") és minél kevésbé szabályozott ezek használata, annál nagyobb lesz az eltérés az írott és az végrehajtott program között. (Kezdő programozó azért szereti a "GO TO" utasítást, mert nagy szabadságot biztosít, oda ugorhat vele, ahová akar. A tapasztalt programozó viszont tudja, hogy feleslegesen nem érdemes ugrálni, ezért csínyján bánik a "GO TO" utasításokkal.)

Nagyon valószínű, hogy amennyiben valamely problémának (feladatnak) van lehetséges megoldása, akkor létezik optimális megoldása is. A kérdés csak az, hogy az optimumot milyen kritériumok alapján kívánjuk meghatározni. Eleinte, amikor a gépek kapacitása meglehetősen korlátos volt, a programvégrehajtási idő és/vagy a program által elfoglalt tárhely minimalizálását tekintették az optimum fő kritériumaiként.

A számítógépek teljesítménynövekedése volt az egyik ok, a megoldandó feladatok méretnövekedése a másik, ami a figyelmet más irányba terelte. A harmadik és egyben döntő ok pedig az az igény volt, hogy a szoftverkészítés „kisipari technológiája" érje utol a hardvergyártás nagyipari technológiáját.

Az „optimum" követelménye ily módon a megírt program érthetőségére, a statikus és dinamikus program valamilyen összhangjára, a futó program „udvariasságára", egyszóval a program „felhasználóbarát" jellegére tevődött át.

programtervezés, strukturált programozás

Ennek az igénynek is köszönhetően a hatvanas évek közepétől alapvető változás ment végbe a programozáselmélet területén is. Elfogadott *programtervezési* módszerré és elvvé vált a *strukturált programozás* logikája, még azon nyelvek esetében is, amelyeknél hiányzik az ehhez szükséges utasításkészlet.

E programozási logika alapja az a matematikailag is bizonyított tétel, hogy minden egy be- és kimenettel rendelkező program leírható három alapvető szerkezet tetszőleges számú egymás utáni, illetve egymásba építésével.

E három szerkezet: a felsorolás (szekvencia), a választás (szelekció), az ismétlés (iteráció).

- Felsorolás például bármely nyelvben az értékadás, a beolvasás és kiírás. A felsorolás viselkedésének lényege, hogy a szerkezetben megadott tevékenységek egymásutáni elvégzését követően a vezérlés kilép a szerkezetből.
- A választás valamilyen feltétel függvényében kétfelé ágaztatja a végrehajtás folyamatát. Ha a feltétel igaz, az egyik ág tevékenységei hajtódnak végre, ha a feltétel hamis, a másik ágon levő tevékenységek. A két ág közül tehát mindig csak az egyik működik. Az egyik ág lehet „üres", vagyis funkciója az, hogy kikerüli a másik ágon levő tevékenységeket. Fontos, hogy a két ágnak közös kilépési pontja van.
- Az ismétlés – mint a neve is mutatja – egy tevékenységsor, esetleg egyetlen tevékenység többszöri (ismételt) végrehajtása. A kijelölt tevékenység(sor) mindaddig ismétlődik, amíg valamilyen feltétel igazzá (esetleg hamissá) válása be nem következik. Ekkor a vezérlés kilép a szerkezetből.

Bizonyítható az is, hogy bármilyen bonyolult a program, a statikus változatból a dinamikus változat az így felépített program esetében lesz legkönnyebben követhető.

Ebből következik a másik fontos programozási elv: igyekezzünk a programozás során ezeket a szerkezeteket létrehozni. Ez mindazon nyelvek esetében könnyű, amelyek ismerik a strukturált programozás szerkezeteinek létrehozását biztosító utasításokat. Azoknál a nyelveknél, amelyek ezeket nem ismerik, vagy nem ismerik az összeset, "GO TO" utasításokkal lehet létrehozni a fenti szerkezeteket.

Ha valaki ezt a két fontos elvet betartja, már nem is érzi azt, hogy nehéz egy jó programot megírni, vagy megérteni. Pedig nehéz!!!

42.

MÓDSZER ÉS CÉL
(A problémamegközelítés problémája)

A számítógép-programozás tanításában gyakran alkalmazott didaktikai fogás, hogy egy-egy probléma elemzése során megkerestetjük a diákkal a probléma matematikai megoldását, majd ennek a megoldásnak a segítségével készíttetjük el a megoldó algoritmust illetve programot.

Sokszor nem is gondolunk arra, hogy a számítógép jelenléte egy, a hagyományos matematikai gondolkodástól eltérő problémaelemzést kíván meg.

Ennek szemléltetésére tekintsük a következő feladatot:

A török szultánhoz 100 fogollyal állít be a nagyvezír. A szultán elrendeli, hogy valamennyit egy-egy zárkába tereljék be. A zárkákon „zárva" és „nyitva" feliratú kallantyúk vannak. A rabok elhelyezése után a szultán végigküldi első szolgáját azzal a paranccsal, hogy valamennyi zárka kallantyúját fordítsa el, azaz zárja be. Majd gondol egyet és elküldi második szolgáját azzal a paranccsal, hogy minden második kallantyún fordítson egyet.

S mivel a szultán ráérő és vicces kedvű ember volt, sorra küldte szolgáit, a harmadikkal minden harmadik, a negyedikkel minden negyedik stb. kallantyút állíttatván az ellenkező állásba. Végül elküldte a századik szolgát, hogy a századik zárka kallantyúját fordítsa el. Ekkor így szólt a szultán: —„Akinek a zárkája nyitva van, hazamehet."

Kérdés:
mely zárkák voltak nyitva, és melyek zárva?

A feladat matematikai elemzése során hamar rá lehet jönni, hogy itt az oszthatóság problémájával állunk szemben. Minden zárkához az a szolga fog hozzányúlni, amelynek sorszáma osztója a zárka sorszámának. Mivel a zárkák kezdetben nyitva voltak, azok maradnak a végén zárva, ahol az osztók száma páratlan, és azok lesznek nyitva, ahol az osztók száma páros. (Az 1 és maga a zárkaszám is osztónak tekintendő.)

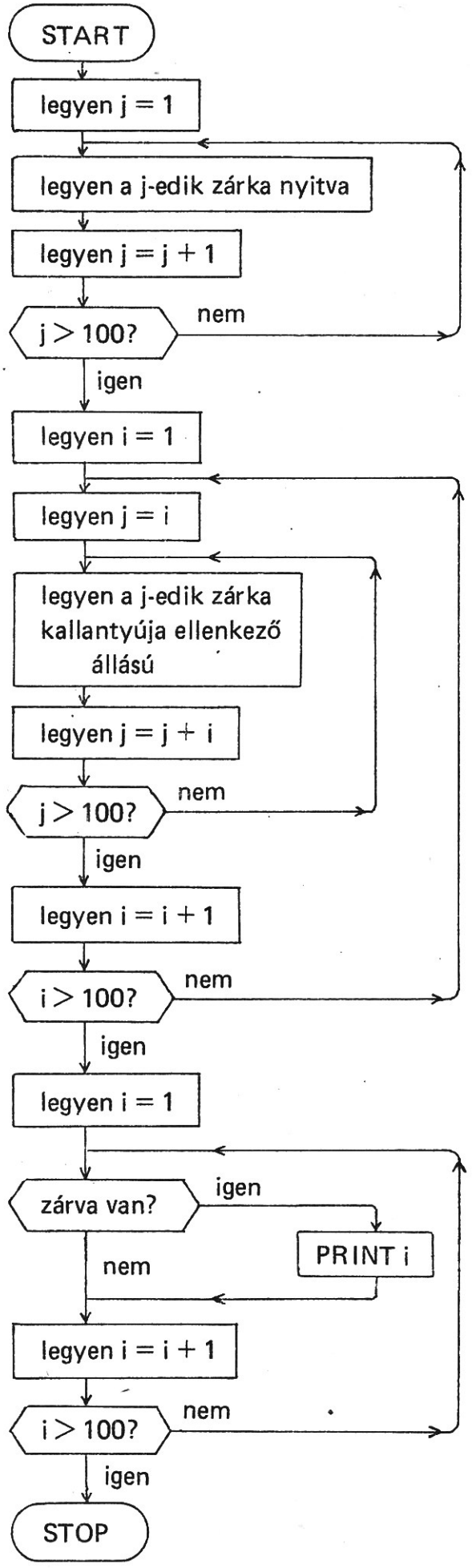
Itt két továbblépési lehetőség adódik. Az egyik szerint 1 és 100 között felírjuk minden szám osztóit, majd megnézzük, hogy melyiké páros, melyiké páratlan. A másik raffináltabb, s azon a megfontoláson alapul, hogy az oszthatóságból következően minden zárkaszám előállítható néhány számpár szorzataként. Melyek azok a számok, ahol a „számpárok" között találunk „fél párt" (hiszen csak így keletkezhet páratlan számú osztó). Nyilvánvaló, hogy ezek a négyzetszámok, ahol tehát a számpár egyik tényezőjének párja önmaga. Akár így, akár úgy okoskodunk, kiderül, hogy azok a zárkák maradnak zárva, amelyeknek sorszáma négyzetszám.

De hogyan oldjuk meg ezt a feladatot számítógéppel?

MÁSKÉPP!

A számítógépes megoldással nem a matematikai modellt, hanem a valóságot modellezzük. A száz kétállapotú kallantyút száz kétállapotú (ha lehet logikai típusú) változóval,

a nyitogatást, csukogatást a logikai változó értékének negálásával (az érték ellenkezőjére állításával) „szimuláljuk". A program terve így néz ki:



(Azért a bezárt zárkák sorszámát írtuk ki, mert abból van kevesebb.)

Ez a programterv bármely nyelven programozható — figyelembe véve az adott nyelv lehetőségeit.

Eredményül az első tíz szám négyzetét (1,4,9,16,25,36,49,64,81,100) kapjuk.

(A programtervet célszerűen grafikus formában, úgynevezett blokksémával ábrázoltuk. Ennek olvasása egyszerű: a ⬭ jel a folyamat kezdetét és végét jelöli, a ⬡ mindig valamilyen döntés vagy vizsgálat, a ▭ pedig művelet vagy esemény. Ezeket az ún. dobozokat folyamatvonal köti össze, amelyen nyílhegy jelöli a történés irányát.)

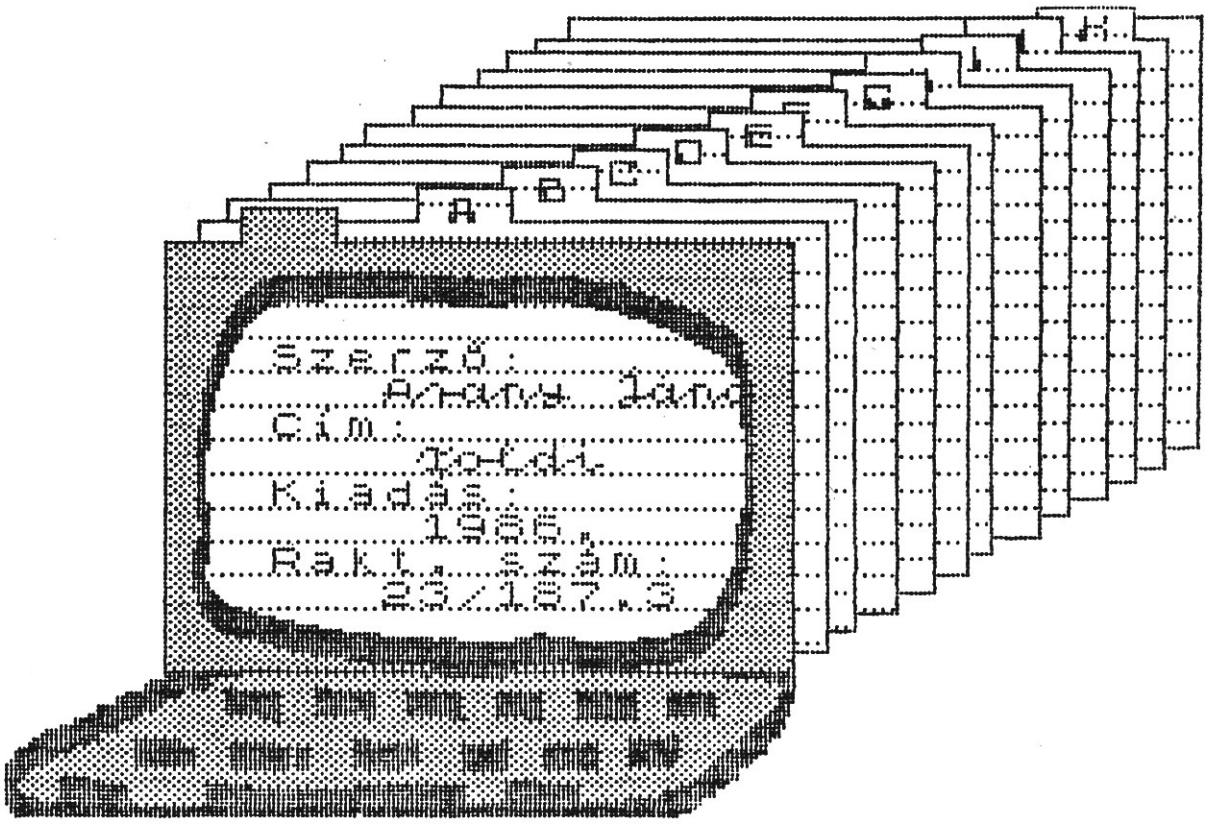
Tanulságok:

- Nem mindig a matematikai modell visz közelebb a számítógépes megoldáshoz.
- Adott esetben ne engedjük magunkat olyan zárkába tuszkolni, amelynek sorszáma négyzetszám.

43.

A REKORDOK ÁLLOMÁNYA
(File, rekord, adatelem)

A számítástechnika és -tudomány szaknyelve az angol. Hiába igyekeznek ezt a nyelv őrei minden országban a saját nyelvükre átalakítani, a szakemberek maguk között továbbra is gyakran az angol kifejezéseket használják, – legföljebb nálunk magyar ragokkal látják el és magyarosan ejtik ki az eredeti angol szavakat. Ennek a „keveréknyelvnek" az elkerülése érdekében adtuk meg a függelékben a legfontosabb számítástechnikai szakkifejezésnek a jelentését és kiejtését.



Ezek közül az adatfeldolgozásban az egyik leggyakrabban használt szó a file (ejtsd: fájl, – melyet egyes szakemberek(?) szerint így is kell helyesen írni. A mi filünknek azonban még – bármennyire is fájlalják – a fájl az file vagy méginkább a magyar megfelelője: *adatállomány*). A kifejezés az angol file szóból származik, mely ebben az esetben kartotékot, iratgyűjtőt jelent és helyes kiejtése körülbelül annyira hasonlít a fájl szóra, mint az angolosan mondott London a magyarosan hangsúlyozottra.

adatállomány

Az *adatállomány* az adatoknak egy meghatározott feladathoz hozzárendelt vagy bizonyos közös tulajdonságokkal rendelkező összessége. A közös állományban levő adatok szerkezete, fizikai tárolási módja azonos. Egy állomány lehet például egy vállalat személyi nyilvántartásában a személyi adatok összessége. Az állomány egyedi információs egységekből, logikailag önálló információt tartalmazó egyedekből, *rekordokból* áll. Az előbbi

példában ezek az egyes személyek adatait tartalmazó „adatlapok", melyek a számítógépes „iratgyűjtőben" természetesen nem papíron, hanem a gép által olvasható mágneses formában vannak tárolva. Minden rekord egy vagy több *elemi mezőből, adatelemből* áll. Az elemi mező a legkisebb, önálló tulajdonsággal rendelkező adat, melyre név szerint lehet hivatkozni. A személyi nyilvántartásban adatelem például a dolgozó neve, beosztása, személyi száma, szakképzettsége. Ezen adatok bármelyikét bármelyik dolgozóra vonatkozóan program segítségével meg tudjuk kapni, vagy meg tudjuk változtatni.

elemi mező, adatelem

Ugyanabban a feldolgozásban több, különböző állományt is felhasználhatunk, melyek felépítése lehet azonos, de eltérő is. Így, ha a személyi nyilvántartást újonnan felvett dolgozók adataival bővítjük, akkor célszerű a már meglevő, valamint az új adatokat külön állományban tárolni. Az előbbit szokták *törzs-*, az utóbbit *változás-* vagy *mozgás-állománynak* nevezni. Ugyanez a rendszer dolgozhat azonban például a vállalat állóeszközeivel vagy termékeivel és azok előállításához szükséges alkatrészekkel, félkész termékekkel is. Ebben az esetben az azonos tulajdonságú egyedek, pl. az összes állóeszköz, vagy az összes beépíthető alkatrész, félkész termék egy-egy önálló állományba kerül.

törzs-állomány, mozgás-állomány

A számítógépeknek nagy tömegű adatfeldolgozási feladatokra való felhasználása szükségszerűvé tette nemcsak az adatok fizikai tárolására, hanem az így tárolt adatoknak hatékony visszakeresésére és módosítására szolgáló módszereknek a kidolgozását is. Ezzel foglalkozik az állomány- (file) szervezés, melyet a következő fejezetben fogunk tárgyalni.

44.

ITT A PIROS – HOL A PIROS?
(Adatszervezési alapfogalmak)

Az adatállományok, file-ok kialakításakor azt szeretnénk elérni, hogy az adataink minél kisebb helyen férjenek el, minél rövidebb idő alatt tudjuk a keresett adatokat megtalálni, valamint, ha újakkal egészítjük ki, vagy meglevőket kívánunk módosítani, akkor azt minél hamarabb végezhessük el. Ezek a követelmények azonban egymásnak többékevésbé ellentmondóak. A technikai lehetőségek (a rendelkezésre álló tároló kapacitása) és a várható felhasználás módja szabja meg, hogy mikor melyik feltételt igyekszünk a legjobban betartani. A gyors visszakeresési és módosítási lehetőség – hacsak nem pontosan úgy, abban a sorrendben keressük, módosítjuk az adatokat, mint ahogyan azok tárolva vannak – további „kisegítő adatok" tárolását teszi szükségessé. Ez újabb tárolóhelyet igényel. Ezenkívül módosításkor, új adatok beírásakor ezeket a megtalálást gyorsító segédadatokat is aktualizálni kell, hogy továbbra is megkönnyítsék, meggyorsítsák a tényleges adatok keresését. Ez megnöveli a módosításhoz, kiegészítéshez szükséges időt, de lerövidíti a keresést. Jó szoftver ellátottságú gépeknél ezen feladatok ellátására kész programok állnak rendelkezésünkre, de más gépeknél nemcsak az adatok szervezési módját kell meghatároznunk és nem csupán az adatokat kell bevinnünk, hanem az ehhez szükséges alapprogramokat is magunknak kell megírnunk. Semmit sem érünk a gyakorlatban egy olyan adatállománnyal, melyhez másodpercek alatt hozzáírhatjuk az új egyedeket, de ha keresünk benne valamit, akkor ezt csak órák múlva találjuk meg, – főleg, hogyha erre másodperceken vagy perceken belül volna szükségünk.

adat-szervezés

Az *adatszervezés* fontosságára és különböző formáira tekintsük a mindannyiunk által jól ismert telefonkönyvet. Száz évvel ezelőtt, amikor egy nagyobb városban is csak néhány tucat telefonelőfizető volt, a telefonkönyv szerkesztése és áttekintése nem okozott különösebb gondot. A kezelőkisasszony fejből kapcsolt minden előfizetőt név szerint, és az egy-két oldalas előfizetői névjegyzékből ránézésre azonnal meg lehetett találni a kívánt nevet és a hozzátartozó számot. Ma, amikor már tíz-, sőt százezres nagyságrendű a helyi előfizetők száma, ez természetesen lehetetlen. Ilyen tömegű adatot már megfelelő módon kell rendszerezni és tárolni, ahhoz hogy a kívánt információt elfogadható időn belül megkaphassuk.

szekvenciális szervezés és elérés

A legegyszerűbb adatszervezési és elérési mód a *szekvenciális szervezés és elérés.* Az információ keresését az adattár elejétől kell elkezdeni és bármely adathoz csak az előtte levők elolvasása után férhetünk hozzá. A magnószalagon levő zene lényegében ilyen egymás után tárolt és visszanyert információk összessége. (A gyors előre- illetve visszatekercselési lehetőség miatt valójában kétféle elérési sebesség van, de ez a soros szervezés lényegét nem érinti.) Egyszerűsége és viszonylagos olcsósága miatt számítógépeknél is igen sokszor alkalmazzák ezt az adatszervezési módot. Az adatokat a magnószalaghoz hasonló mágnesszalagon tárolják és felírásuk illetve leolvasásuk is a magnófelvételhez illetve visszajátszáshoz hasonló módon történik. Természetesen a zenei hangok helyett a számítógép számára értelmezhető jeleket használnak.

A nagyszámítógépekhez csatlakoztatott mágnesszalagokon centiméterenként általában 300–2400 jelet (betűt, számot vagy egyéb írásjelet) lehet tárolni. Egy szabványos tekercsre mintegy 15–20 millió jel vihető fel, így az egész budapesti távbeszélőnévsor bőven ráfér egy mágnesszalagra. Ha a teljes telefonkönyvet erről a mágnesszalagról kellene rendszeresen kinyomtatni, akkor ez az adatszervezési mód lenne a leggazdaságosabb. Az adatok már eleve abban a sorrendben kerülnek feldolgozásra, ahogy eredetileg tároltuk őket. Ugyanakkor viszont, ha egy ilyen sorosan szervezett adattárból véletlenszerűen kiválasztott, különböző helyen levő adatok iránt érdeklődünk, akkor ezek megtalálása meglehetősen nehézkes.

Képzeljük el, mi volna, ha pl. Tóth Ferenc telefonszámát csak úgy tudnánk megtalálni, hogy Aba Botondtól kezdődően minden egyes nevet és telefonszámot végigolvasnánk a telefonkönyvben, míg el nem jutunk a keresett személyig. Ez még az emberi szemnél sokkal gyorsabban olvasó számítógépnek is átlagosan több percig tartana, ami nem olcsó mulatság. Éppen ezért a szekvenciális szervezésű adattárakat olyankor használják, amikor egy vizsgálatnál az összes adatra vagy az adatok nagy részére szükség van, mégpedig ugyanabban a sorrendben, ahogy azokat eredetileg is eltárolták. Ha az információkat a szekvenciális adattárból nemcsak a tárolási sorrendben kívánjuk megkapni, akkor lényegesen meggyorsítja a keresést az indexek alkalmazása. Ezt az adatszervezési módot *indexelt-szekvenciális* szervezésnek nevezzük. Lényegében ilyen elven épül fel a telefonkönyv is. Az első index a könyv oldalán levő betű, amely mutatja, hogy az ezzel a betűvel kezdődő nevek melyik oldalon találhatók. Mivel ez jól látható módon elkülönül a tényleges információtól, a nevektől és a telefonszámoktól, ezért Tóth Ferenc telefonszámának keresésekor a könyvet azonnal ki tudjuk nyitni a T betűnél, és így legrosszabb esetben csak a T-vel kezdődő neveket kell sorban végigolvasni.

indexelt-szekvenciális elérési mód

Mivel azonban ezek száma is meglehetősen nagy lehet, ezért a keresés meggyorsítására még egy index, a lapindex szolgál.

Ez is elkülönül a tényleges információtól, mindig a lap tetején található, és megadja az azon a lapon levő utolsó (és/vagy első) nevet. Ezért a keresést a T betűs lapokon célszerű úgy folytatnunk, hogy a teljes névsor helyett csupán ezt a lapindexet nézzük, és amikor ennél eljutottunk a Tóth névig vagy túlhaladtunk rajta, akkor kezdjük el az ehhez tartozó oldalakon a nevek sorba való végigolvasását.

Az indexelt-szekvenciális adatszervezési módot rendszerint akkor alkalmazzák, amikor az adatokat mind fizikai sorrendben, mind pedig rendezetlen, véletlenszerűen változó sorrendben akarják elérni. Ilyen lehet például egy vállalati személyi nyilvántartás, ahol fizetéskor sorban minden dolgozó adatát fel kell dolgozni, de gyakran kell információt szolgáltatni egyes kiválasztott személyekről is. Természetesen ennek a szervezési és hozzáférési módnak alapfeltétele, hogy az adatok a kívánt információ szerint sorba legyenek rendezve. A telefonkönyv példájánál maradva, semmit sem érnek a néven alapuló indexek akkor, ha pl. egy előfizetőt a lakcíme vagy a telefonszáma alapján kívánnánk azonosítani.

Ahhoz, hogy az indexek alapján gyorsan a megfelelő adatcsoporthoz jussunk, olyan adattároló eszközökre van szükségünk, melyeknek egyes helyei megcímezhetők, és az érdektelen adatokon való áthaladás sokkal gyorsabb az adatok elolvasásánál. Számítógépeknél erre a célra legjobban a mágneslemezek váltak be, melyekről a 22. és 23. fejezetben írtunk részletesen.

Abban az esetben, ha az adatokat csak véletlenszerűen megadott sorrendben kívánjuk feldolgozni, akkor leggazdaságosabb úgy tárolni őket a mágneslemezen, hogy a keresésre szolgáló ismérvből azonnal megkaphassuk tárolási címüket, azt, hogy melyik cilinder,

melyik sávján vannak. Ezt *közvetlen* vagy *direkt szervezési és elérési módnak* nevezzük. Ez a szervezés egyszerűen megvalósítható például az előfizetői adatoknak telefonszám alapján való visszakeresésénél, ahol a telefonszámból valamilyen matematikai algoritmussal megkaphatjuk a cilindert, illetve sávot, ahol az adott számhoz tartozó további információk megtalálhatók. Tulajdonképpen egy direkt keresést valósít meg a függelékben levő betűrendes tárgymutató is. Itt a fogalom után levő számok adják meg az ismertető anyag tárolási címét, vagyis hogy az melyik fejezetben, melyik oldalon található. A számítógépet működtető program ráállítja a kiszámított, megadott cilinderre és sávra az író-olvasó fejet és a kívánt információ beolvasható innen illetve beírható oda (az olvasó a megfelelő oldalra lapoz).

közvetlen szervezési és elérési mód

Tovább bonyolítja a helyzetet, hogyha az adatokat különféle ismérvek alapján is vissza kívánjuk keresni. Például egy vállalati nyilvántartásban a dolgozókat nevük alapján ábécé sorrendbe vettük fel egy indexelt-szekvenciális adattárba. Ebben az esetben a neve alapján bármelyik dolgozó adatait meg tudjuk találni. Ha azonban azt kívánjuk megtudni, hogy kik azok a dolgozók, akik az utolsó félévben fizetésemelést kaptak, akkor erre csak úgy válaszolhatunk, hogy végigolvassuk az összes dolgozó erre vonatkozó adatát és ezekből kigyűjtjük azokat, akik a kérdéses feltételnek megfeleltek, vagy eleve elkészítünk az eredeti adattártól függetlenül egy másik adattárat, mely csak a fizetésemelést kapott dolgozók adatait tartalmazza. Az előbbi esetben igen sok adatot fölöslegesen olvasunk be (főleg akkor, ha a dolgozóknak csak egy kis része kapott fél éven belül fizetésemelést), míg a másodikban fölöslegesen kétszer tároljuk ugyanazon adatokat. Ez főleg akkor okoz gondot, ha sok dolgozó kapott emelést, mivel nagy adatfeldolgozási feladatoknál sokszor még a számítógépek első pillantásra óriásinak tűnő tároló képessége is gyakran szűknek bizonyul. Az ilyen többszörös tárolásnál ezenkívül ügyelnünk kell arra is, hogyha valamelyik adaton változtatást hajtottunk végre, akkor azt a többi adatállományban elhelyezkedő összes másolatán is következetesen véghezvigyük.

A több ismérv alapján való visszakeresés problémájának egyik megoldása az *invertált állomány.* Itt egy olyan segédfile-t, más néven *indexállományt* hozunk létre, melynek egyedeit a különböző ismérvek alapján tudjuk megtalálni és ezek az adott értékhez tartozó adatok tényleges címeit tartalmazzák. Ha pl. két ismérv együttes megléte alapján kell a megfelelő adatrekordokat kiválasztani, akkor azok felelnek meg ezen együttes feltételeknek, melyek mindkét ismérv címlistáján rajta vannak.

invertált állomány, index-állomány

A fentiek könnyebb megértésére tegyük fel, hogy a 32 lapos magyar kártya lapjait tároljuk egy 32 tárolóhelyes számítógépben és a lapokat szín, nagyság és mindkettő alapján meg kell tudnunk találni. Ekkor a következő két segédfile-t építjük fel a kártyák elhelyezésénél:

Indexállomány nagyság szerint						**Indexállomány szín szerint**							
nagyság	**tárolási hely**				**szín**	**tárolási hely**							
VII	4	8	16	27	piros	1	2	14	16	18	19	27	31
VIII	1	3	9	17	zöld	3	5	12	20	21	22	28	32
IX	16	18	21	30	makk					.			
X	7	12	13	29	tök					.			
...	...	...	...	...	...	...	...	...	...	...	...	...	...

A piros hetes megtalálásához nem kell az adatokat mind végignézni. Elég összevetni a szín indexállomány piros sorának címeit a nagyság indexállomány VII sorának címeivel,

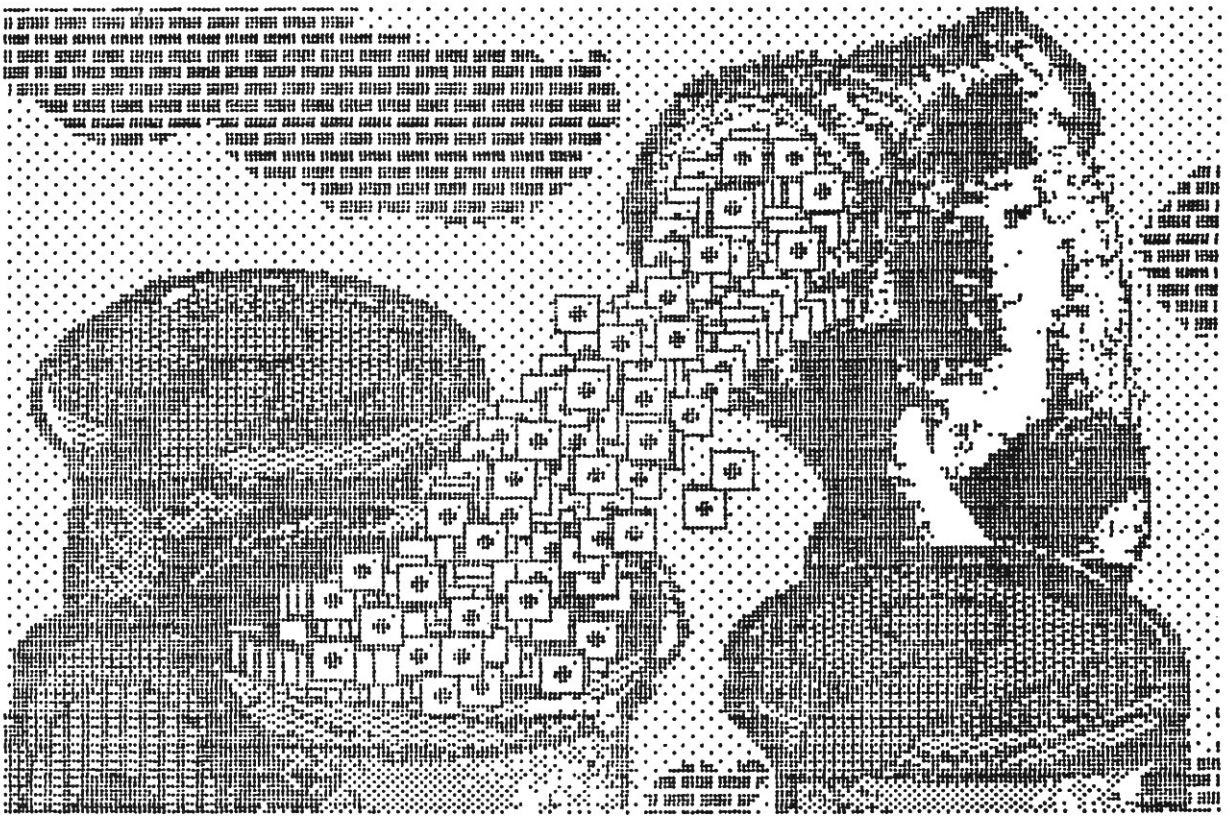
hogy megtaláljuk az egyetlen közös címet, a 16-ot. Itt van a piros hetes! Ha ezzel megértettük az invertált állomány lényegét, akkor nyertünk! Ha viszont nem találtunk közös címet, vagy egynél többet találtunk, akkor hibásan építettük fel az indextáblát, vagy hamis kártyapaklit használtunk.

45.

HA SZEMÉT MEGY BE, SZEMÉT IS JÖN KI
(Adatrögzítés, adatellenőrzés)

A számítógépekkel foglalkozó könyvekben, cikkekben sok szó esik a gépek csodálatos sebességéről, óriási tárolókapacitásáról, műszaki paramétereiről, a rendelkezésre álló szoftverről, az alkalmazási területek sokrétűségéről, de viszonylag keveset hallhatunk arról a fáradságos munkáról, ami a feldolgozandó adatoknak a gép számára értelmezhető formába való átalakításához szükséges. Az *adatrögzítés* szerény Hamupipőkeként húzódik meg a látványosabb munkafolyamatok mögött, holott adatok nélkül egyetlen számítógép sem tud hasznosan dolgozni. Az adatrögzítés fontosságát támasztja alá az is, hogy a számítógépek fő alkalmazási területe ma az adatfeldolgozás, ahol a legnagyobb hangsúly éppen a nagy adattömeg gyors és pontos feldolgozásán van.

adat-rögzítés



A szokványos adatfeldolgozási munkáknál a feladat elvégzéséhez szükséges adatok összegyűjtési és rögzítési ideje általában többszöröse a tényleges feldolgozásukhoz szükséges időnek. Már egy közepes nagyságú számítógépre is igaz, hogy egy csupán néhány óra gépidőt igénybe vevő ügyviteli, üzleti jellegű feladatnál is néhány tucat adatrögzítő több napos munkája szükséges ahhoz, hogy az eredeti alapbizonylatok tartalmát terminálokon keresztül közvetlenül bevigyék a gépbe, vagy a gép által olvasható formába, mágnesszalagokra, mágneslemezekre írják.

Az adatrögzítési munka nemcsak hogy sok időt igényel, hanem nagy figyelmet is követel. Gyakorlott adatrögzítők attól függően, hogy csak számokból vagy számokból és betűkből is álló adatokat kell bevinniük a gépbe, óránként 6–10 ezer jel leütésére is képesek. Sajnos azonban ezek nem biztos, hogy mind hibátlanok. Előfordulhat, hogy az eredeti, a bizonylaton levő adatok hibásak, de tévedhet az adatrögzítő is. A gyakorlat azt mutatja, hogy jó munkaerőnél ezen utóbbi hibák aránya 0,05–1 ezrelék körül van, azaz minden ezredik–húszezredik jel hibás. Ez akármilyen kedvezőnek is tűnik az első pillanatban, azt jelenti, hogy például egy ötjegyű számokból álló fizetési jegyzéken egy 4000 fős vállalatnál átlagban 1–20 személynél minden hónapban hibás összeg szerepelne a fizetési jegyzéken. Természetesen a reklamációk száma ennél kevesebb lenne, hiszen valószínűleg csak azok szólnának, akik kevesebb pénzt kaptak.

A fentiek alapján érzékelhetjük, mennyire fontos, hogy a számítógépnek a munka elkezdésekor megadott kiinduló adatok száz százalékosan hibátlanok legyenek. A hibák kiküszöbölésének, illetve minimálisra csökkentésének az egyik leggyakoribb módja, hogy az adatokat (legalábbis azokat, melyeknek helyességét más összefüggések alapján nem tudjuk ellenőrizni) két személy viszi be a gépbe egymástól függetlenül. A program csak akkor fogadja el az adatokat, ha mindketten pontosan ugyanazt rögzítették. Eltérés esetén ellenőrizni kell, melyik volt a jó.

Egy ügyes háziasszony rossz alapanyagokból is tud olyan ebédet főzni és azt úgy feltálalni, hogy mindenki jóízűen egyen belőle. Egy számítógép képtelen hasonló bűvészmutatványra. Ha az alapanyagok rosszak, akkor azokból a legjobb program, a legnagyobb teljesítményű gép sem fog helyes eredményeket, megalapozott döntést előkészítő javaslatot kihozni. Vagy ahogyan a számítástechnikusok mondják: ha szemét megy be, szemét is jön ki. Éppen ezért minden számítógépes eredményszolgáltatás megbízhatóságának alapja a bemenő adatok hibátlansága.

Viszonylag kicsi a hibalehetőség, ha az adatok egy másik, már ellenőrzött számítógépes rendszerből, vagy közvetlenül műszerekből jönnek a gép által azonnal feldolgozható formában, például mágnesszalagon vagy gép-gép közti közvetlen kapcsolattal. A fő hibaforrás az ember, aki hibásan adja meg az adatokat. Akár véletlenül, mert rosszul olvas el egy jelet, akár szándékosan, mert például egy hölgy letagad az életkorából. Az ilyen hibát elsősorban az adatok felvételénél, az utóbbi példában esetleg közvetlenül ránézéssel kell kiszúrni, illetve kiszűrni. Minden adatfeldolgozó rendszernek alapkövetelménye, hogy a felhasznált adatokat közvetlenül keletkezésük helyén gondosan ellenőrizni kell. Az egyes tételek egyeztetésén kívül a bemenő adatok hibáinak felderítésére szolgál az is, hogy sokszor úgy kérik azokat, hogy több adat között valamilyen logikai kapcsolat álljon fent.

A gép ellenőrzi, hogy ez teljesül-e. Ha nem, akkor visszaküldi az adatszolgáltatónak, hogy javítsa ki a nem megfelelő értékeket.

Igen egyszerű, de a gyakorlatban sok hibát kiszűrő ellenőrzési mód, hogy a programban számoljuk a feldolgozott tételek számát és ezt összevetjük a beolvasandó bizonylatok számával. Sokszor kiderül ebből, hogy egy-két bizonylat nem érkezett el a géphez, valahol menet közben elkallódott.

Az adatfelvételre szolgáló kérdőívekre sokszor szóbeli választ kell adni. Az összehasonlítás egyszerűbbé és egyértelműbbé tétele érdekében ezeket a válaszokat egy-két jelből álló rövidítéssel kódolják és ennek értékét viszik be a gépbe. Azt a program ellenőrizni tudja, hogy ez a kód a ténylegesen létező kódok valamelyike. De azt, hogy az valóban a helyes érték is, csak akkor, ha az adatok között más összefüggés is fennáll.

Igen sokszor a számítógépes rendszerben szereplő egyes egyedeket számokkal azonosítják. Ezek a számok rendszerint bizonyos fokig „beszélnek" is, meghatározott helyen levő jegyek értéke az egyed valamilyen tulajdonságára utal. Például egy raktár nyilvántartásában minden egyes cikkelemet egy egyedi szám azonosít, amelyiknek első jegyei azt az árucsoportot jelzik, amelyikbe tartozik, vagy a személyi számnak első száma az illető nemét, további 6 számjegye pedig születési időpontját jelzi. Ilyen hosszú számok rögzítésénél azonban óhatatlanul előfordulhatnak hibák. Ennek kiszűrésére a már önmagában is azonosító számsorozat mögé tesznek még egy vagy két további jegyet (ellenőrző számot) úgy, hogy az így kapott azonosító eleget tegyen valamilyen matematikailag jól meghatározható feltételnek. Széles körben elterjedt a 11-es *ellenőrző szám.* Ennek egyszerűbb változatában az ellenőrző számok(ka)t úgy választják meg, hogy a számjegyek összege 11-gyel osztható legyen. Ez a módszer egy-egy számjegy hibás rögzítését biztosan kijelzi, általában kettőét is, de azt a gyakori hibát, amikor két jegyet fölcseréltek, már nem.

ellenőrző szám

Nagyobb védelmet ad, amikor az egyes számjegyeknek a helyértéktől függő tényezővel megszorzott összege kell, hogy osztható legyen 11-gyel. Így ellenőrzik a személyi számokat is, mint azt az 52. fejezetben ismertetjük. Ez az eljárás nemcsak bármely szám hibás beütését, de bármely két szám felcserélését, sőt a legtöbb esetben bármely két szám rossz beütését is jelzi. Ebben az esetben tehát igaz a fociszabály általánosítása:

Tizenegyesnél csak rosszul lehet rúgni – rosszul ütni nem.

46.

KÉTSZER KETTŐ NÉHA ÖT
(Műveletek kerekített számokkal)

Nincsen új a nap alatt, kétszer kettő az mindig négy. Alapvető közhely ez, — és mint minden ilyen — csak fél igazság.

Minden, matematikailag kissé tájékozott ember előtt ismeretes, hogyha kerekített számokkal dolgozunk, akkor kétszer kettő eredménye kettő és hat, míg kettő meg kettő az három és öt között változhat attól függően, hogy a kettőt például 1,5 és 2,4 között minek a kerekítéséből kaptuk:

$$1{,}5 \times 1{,}5 = 2{,}25 \rightarrow 2;\ 2{,}4 \times 2{,}4 = 5{,}76 \rightarrow 6;$$
$$1{,}5 + 1{,}5 = 3;\ 2{,}4 + 2{,}4 = 4{,}8 \rightarrow 5 \text{ (kerekítve)}$$

Hiába tudjuk azonban ezt, a tökéletes számszaki egyezéshez hozzászokott pénzügyi, könyvelői szemlélet mindig megborzong, és hibát gyanít, ha 2+2 nem pontosan 4, ha a szükséges ellenőrző összegek, és kötelező számszaki összefüggések nem egyeznek meg a kimutatásokban közölt utolsó jegyig is.

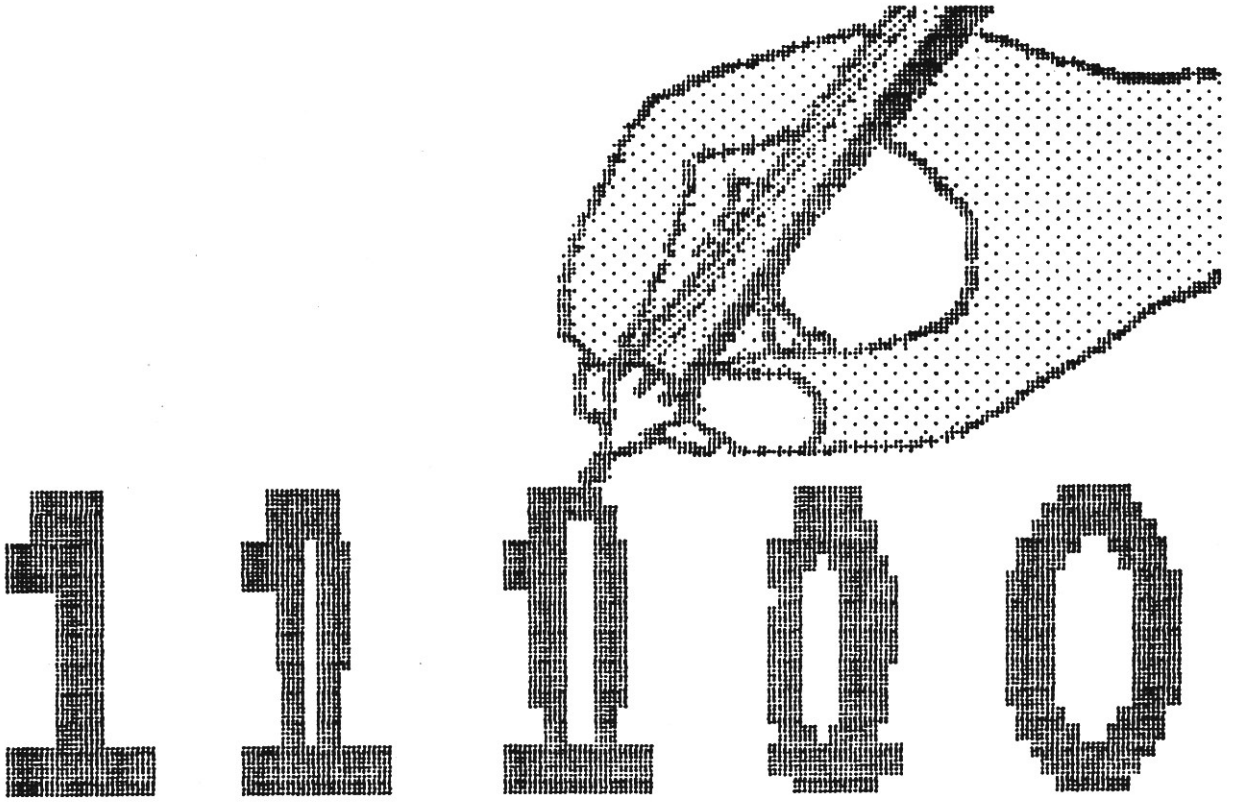
Talán csak P. Howard híres regényhőséhez, Gorcsev Ivánhoz — aki a makaó nevű tiltott szerencsejátékban nyerte el a Nobel-díjat — hasonlítható az a matematikus, aki abból gazdagodott meg, hogy egy nagy bank számítógépes rendszeréhez olyan programot készített, mely az ügyfél helyett való kifizetésnél mindig fel, befizetéskor pedig lekerekítette a centeket dollárra. A néhány centes eltérés miatt senki sem reklamált. A belső elszámolást azonban a gép centre pontosan végezte, és a pénztárossal összejátszva a kétféle elszámolásból adódó különbözetet szétosztották. Valószínű, hogy az a történet nem igaz — a felvett pénzből talán egy 14 karátos kerékpárra sem telt volna —, de mindenesetre megtörténhetett volna, — és mutatja a harmadik örök igazságot: sok kicsi sokra megy.

A fillérre egyező összesítéseknek természetesen megvan a létjogosultsága. Akkor, amikor egy műveletnél sem használunk kerekített tényezőket, minden ellenőrző összefüggésnek pontosan kell teljesülni. A hagyományos kézi feldolgozásnál a pontos egyezés egyben a számolás bizonyos fokú ellenőrzésére is szolgál. Számítógépeknél erre nincsen szükség, a gép maga ellenőrzi saját magát.

Ugyanakkor viszont a számítógép eleve véges hosszúságú számokkal számol, vagy azért, mert a műveletvégző egységébe nem fér be több számjegy, vagy azért, mert a megoldandó feladat sem követeli meg a tökéletes pontosságot.

Ez utóbbira lássunk is egy példát. Hiába tartalmaznak az egyes tételek igen sok számjegyet, magasabb szinten ebből rendszerint csak az első néhány jegy bír jelentőséggel. A könyvelő számára talán még fontos lehet, hogy egy üzemrész napi anyagszükséglete 5627 Ft 12 fillér, vagy 5627 Ft 79 fillér volt, de az igazgató számára a napi jelentésben 5627 Ft, sőt általában 5,6 ezer Ft is elegendő információt tartalmaz. A felsőbb hatóságok felé a teljes gyárra vonatkozó negyedéves adatszolgáltatásnál pedig rendszerint elegendő megadni az összértéket, pl. millió forintban, 2 tizedessel. Nem kell különösebben magyarázni, hogy az így összeállított, kevesebb számjegyet tartalmazó táblázat mennyivel áttekinthetőbb és ugyanakkor a vezetők számára az összes értékes információt tartalmazza.

Számítógépes feldolgozásnál is előfordulhat, hogy kerekített mennyiségekkel közvetlenül végzett műveletek eredménye nem egyezik meg a kerekítetlen adatokból számolt és helyesen kerekített végeredménnyel. Ha a gép a kerekítetlen, vagy legalábbis több jegy pontossággal megadott adatokkal számolt, úgy ez mindig pontosabb, mint a kerekített számokkal végzett műveletek eredménye. Az eltérésen ugyanúgy nem szabad csodálkoznunk, mint amikor az országúton azt látjuk, hogy Bugyi 16 km, Sári 28 km, majd néhány kilométerrel odább azt, hogy 13, illetve 24 km. Nem Sári került közelebb Bugyihoz, csupán a kilométerre történő kerekítések miatt változott meg látszólag a két helység távolsága.



És hogy egy 100 százalékosan ideillő példát is bemutassunk, nézzünk egy jellegzetes mintát a statisztika világából. (Itt már eleve bizalmatlanul szemlélhetjük az eredményeket. Tudjuk, hogy a statisztikával mindent és még annak az ellenkezőjét is be lehet bizonyítani, csak megfelelően kell a statisztikai mintavételt elvégezni. Így például egyik kis tanítványunknak alig bírtuk elmagyarázni, hogy lehet igaz a statisztika, miszerint a családok mintegy 10 százaléka gyerektelen, amikor az ő több száz családot képviselő iskolatársai körében egyetlen egy ilyennel nem találkozott.)

Tegyük fel, hogy reggel minden harmadik ember tejet, minden harmadik teát, minden harmadik kávét iszik. (A mintavétel, a felmérés 9 óra előtt történt, amikor még tilos az alkoholtartalmú italok árusítása.) A statisztika szerint az egyes italok százalékos megoszlása 33; 33; 33 (összesen 100 és nem 99!) vagy 33,3; 33,3; 33,3 (összesen 100,0 és nem 99,9!) százalék. Ha egy újság a fenti adatokat közölné, biztos kapna néhány félművelt embertől kioktató olvasói levelet vagy más újságban gunyoros glosszát, megkérdezvén, mi történt a hiányzó 1 vagy 0,1 százalékkal. Semmi! Elveszett a kerekítésnél. Ennél több is elveszett Mohácsnál. Még mindig helyesebb azt írni, hogy összesen 100 százalék, mert a hozzáértő tudja, hogy az eltérés a kerekítésből adódik, mint akármelyik értéket önkényesen megnövelni 34 vagy 33,4-re, illetve az összegekbe a látszólag helyesnek tűnő 99 vagy 99,9-et írni. Ez utóbbi esetben ugyanis joggal lehet arra következtetni, hogy a hiányzó 1 vagy 0,1 százalék semmit sem iszik, vagy zugivó.

47.

ÍGY PROGRAMOZUNK MI (?)
(Milyen a jó program és a jó programozó)

Egy magasugrónak vagy kalapácsvetőnek a teljesítménye könnyen mérhető. Az a felnőtt, aki átugorja a 220 centimétert, az jó magasugró, aki csak 140-et, az nem; aki 80 méterre hajítja a kalapácsot, az kiváló kalapácsvető, aki csupán 40 m-re, az bizony igen gyenge. Nem ilyen egyértelmű már a megítélés a pontozásos sportágakban. Ha egy ökölvívót kiüt az ellenfele, akkor nyilvánvaló, hogy ki a győztes, de ha a pontozóbíráknak kell dönteni, akkor bizony gyakran ők maguk is különböző véleményen vannak és a közönség sem ért mindig egyet velük.

Még nehezebb egy számítógépes programozó munkájának megítélése. Nem csupán azért, mert szakmájához sokkal kevesebben értenek, mint a sporthoz (különben is, a szakismeret hiánya nem mindig kizáró ok a bírálatban!), hanem azért is, mert a vele szemben támasztott követelmények sokkal szerteágazóbbak. A bevezetőben leírt példánál maradva, igazságtalanság volna azt mondani a 220 cm-t átvivő ugróról, hogy rossz atléta, mert csupán 40 m-re tudja eldobni a kalapácsot.

Mindenkinek az eredményességét azon a területen kell vizsgálnunk, amelyiken ő maga eredményt akar felmutatni, vagy amelyiken a közösségnek szüksége van a képességeire és teljesítményére. Igen leegyszerűsített az a személet, hogyha egy program pontosan azt végzi el, amit kívánnak tőle, akkor az jó program, és az a programozó, aki ezt a kívánt határidőre el tudta készíteni, az egy jó programozó.

A hivatásos programozó (nem a szervező-programozó) munkája sok szempontból egy fordító munkájához hasonlítható. Egy adott nyelven (emberi nyelven) megfogalmazott feladatot át kell hűen fordítani egy másik nyelvre. Természetesen a fordítók között is vannak különbségek. Vannak műfordítók, akik saját egyéniségüket is hozzáteszik a fordításhoz és ezáltal az még jobbá válik az eredetinél (példa erre Karinthy Frigyes Micimackó fordítása), de vannak, akik „csupán" hűen visszaadják az eredeti szöveg értelmét, de hangulatát, ritmusát, stílusát már nem.

A programozásnál is sokféle szempont van, melyet a jó programozónak figyelembe kell venni. Természetesen alapkövetelmény, hogy a program pontosan azt hajtsa végre, amire készült, ami a szervezési dokumentációban szerepelt. Ezt az eredményt azonban sokféle módon el lehet érni.

Érdekes kísérleteket végeztek programozók munkájának összehasonlítására. Hogy az egyéni képességekből és tulajdonságokból adódó eltéréseket kiküszöböljék, több főből álló programozói csoporttal készíttették el ugyanazokat a feladatokat. Az egyik társulat programja például 1966 bájtot foglal le, míg a másiké (ugyanarra a feladatra!) 13 438-at, az egyik 33 PL/1 utasításból állt, a másik 166-ból, volt olyan csoport, amelyik 24 embernapi munkával készítette el a programot, volt amelyik 74-gyel.

Lehet-e ezekből a számokból egyértelmű következtetéseket levonni az egyes csoportok teljesítményére vonatkozóan? Bármennyire is meglepőnek tűnik első pillanatra, de nem lehet. Minden programozónak megvan a saját munkamódszere, lelkivilága (és hogy ez mennyire érzékeny, azt csupán a programozási osztályok és számítóközpontok vezetői tudják) és ha nem fogalmazzák meg pontosan azt, hogy a program azon kívül, hogy helyes

eredményeket szolgáltat, még milyen mellékfeltételeket elégítsen ki, mindegyikük saját stílusának megfelelően készíti azt el.

Az egyik kísérletben egy 20 oldalas specifikációban csupán egy mondat utalt rá, hogy az egyik csoportnak a lehető leghamarabb, s legkevesebb próbálkozással kell a végleges programot elkészíteni, míg a másiknak az volt a célja, hogy minimális futási idejű programot állítson elő. Az eredmény megdöbbentő volt. Az első csoport 29 futtatás után szolgáltatta a hibátlan, minden szempontból ellenőrzött programot, míg a másodiknak csak 69 próbálkozás után sikerült ugyanaz. Ugyanakkor viszont a már elkészült programok közül az első futási ideje hatszor hosszabb volt, mint a másodiké.

Egyszerűen leszűrhető ebből a tanulság: Sok esetben csupán azzal, hogy megmondják a programozóknak, milyen szempontból kell optimalizálni a programjaikat, hasonló eredmény érhető el, mint hogyha drága pénzért külön hardver- vagy szoftverkiegészítést vásárolnának a géphez.

A legfontosabb — egymással azonban sajnos teljesen ellentétes — szempontok, amelyeket egy program optimalizálásakor figyelembe kell venni, a következők:

— minimális központitár-igény,
— minimális futási idő,
— legrövidebb elkészülési idő,
— könnyű változtathatóság.

Ezeken kívül további optimalizálási szempontok lehetnek még:

— minimális perifériaigény,
— minimális utasításszám,
— az eredmények áttekinthetősége,
— a program áttekinthetősége.

Hogy ezen szempontok közül melyiket vagy melyeket kell a legfontosabbnak tekinteni, az mindig az adott feladattól és konfigurációtól függ.

Programozói csoportok optimalizálási szempontjai és azonos programok elkészítésénél a különböző szempontok szerint elért rangsorolások.

Szempont	Az adott szempont szerint elért sorrend				
	A	B	C	D	E
A	1	4	4	2	5
B	5	1	1–2	5	2–3
C	3	2	1–2	3	4
D	2	5	3	1	2–3
E	4	3	5	4	1

A = minimális helyfoglalás a tárban
B = eredmények áttekinthetősége
C = a program áttekinthetősége
D = minimális utasításszám
E = legrövidebb elkészítési idő

Kísérleteket végeztek programozói csoportokkal annak megállapítására, hogy menynyire sikerül egy előre megadott szempont szerint optimalizálni programjukat és hogy ez mennyiben befolyásolja az egyéb szempontok kielégítését. Az eredményt az előbbi táblázatban foglaltuk össze. Jól látható, hogy minden egyes csoport első helyen végzett a számára megadott szempont szerinti rangsorolásban. Ugyanakkor viszont a leggyorsabban dolgozó csoport gyengén szerepelt az összes többi tényező szerinti értékelésben. Érthetően szoros korreláció volt a minimális tárolóigény és a legkevesebb utasítás, valamint a program és az eredmények áttekinthetősége között. Ezzel egyidőben ellentétes az eredmények áttekinthetősége és a minimális utasításszám és tárolóigény, ami nyilvánvaló, hiszen a jól formázott eredménykiírás sok átviteli utasítást és ezáltal tárolóhelyet igényel.

Ezekből a tapasztalatokból egyértelműen következik, hogy egy program és egy programozó jóságát csak annak alapján szabad megítélnünk, mennyire közelíti meg megoldása a megadott szempontok szerinti követelményeket. Értelmetlen azt mondani, hogy pl. egy 200 kbájt helyfoglalású fordítóprogram sokkal jobb, mint a 8 kbájtos, mert az utóbbi lényegesen lassabb, és igen sok megszorítást tartalmaz. Akinek nincsen megfelelő nagyságú gépe, az nem is tudja használni a nagyobb, „jobb" fordítóprogramot, számára az nem csupán rossz, de használhatatlan.

48.

SZABADÍTS MEG A GONOSZTÓL!
(Hibakeresés számítógépes programokban)

Annak ellenére, hogy az élet kegyetlenül bebizonyítja az ellenkezőjét, számos naív programozó, sőt időnként talán e könyv írói is bíznak abban, hogy programjuk rögtön, az első változat megírása után hibátlanul működik. Előbb-utóbb azonban mindannyian szomorúan tudomásul vehetjük a keserű igazságot, miszerint a programkészítés göröngyös útján számos probléma merül fel, melyeket csak fokozatosan, lépésenként lehet helyesen megoldani.

A programozás egyik legfárasztóbb, gyakran legnagyobb szellemi erőfeszítést igénylő része egy program hibátlanságának ellenőrzése, illetve a benne levő hibák megkeresése és kijavítása. Hogy ez általában mennyire népszerűtlen munka a programozók körében, azt mutatja a hibakeresés folyamatának angol elnevezése: „program debugging", amit magyarra – érzékeltetve a kifejezés hangulatát is – legjobban talán a „program tetvetlenítése"-ként fordíthatunk.

Nyomon vagyunk!

Ez a legtöbb esetben időt fecsérlő, többszörös nekifutást, próbálkozást igénylő folyamat. Mégis a legélvezetesebb tevékenység lehet, ha valamelyik nem túl kedvelt kollégánk programjában veszünk észre egy olyan hibát, amit szerzője képtelen volt megtalálni. A gyakorlatban a következő hibafajtákkal számolhatunk:

hardver-hibák

Hardver-hibák. Ezek szerencsére ritkán fordulnak elő. Jó számítógép üzembiztonsága csaknem 100%. Ezenkívül a jelenlegi műszaki fejlettség következtében a rendszer majdnem minden ilyen jellegű hibát automatikusan kijelez. (Lásd a 17. fejezetet!)

rendszer-szoftver-hibák, üzemeltetési hibák, rendszer-tervezési hibák, programozási hibák

Rendszerszoftver-hibák sem igen adódhatnak, ha a gyári vagy a nagy szoftverkészítő cégek sokszorosan ellenőrzött szoftverjeit használjuk módosítás nélkül. (Hacsak nem alkalmazzuk rosszul a rendszert!)

Üzemeltetési hibák, mint például nem megfelelő adathordozó föltétele, nem megengedett bemenő adatok megadása, jól megtervezett felhasználói ellenőrző programokkal és a rendszerszoftver ellenőrző funkcióinak kihasználásával könnyen elkerülhetők, illetve kijavíthatók.

A *rendszertervezési hibák* megállapítása kényes kérdés. Nem könnyű éles határvonalat húzni a rendszertervező és a programozó munkája között, meddig terheli a rendszertervezőt a felelősség és honnan kezdődik a programozóé. Kivétel ez alól, ha a kettő egy személy, mert ekkor a felelősség egyértelműen az övé.

A könyv olvasói leggyakrabban a *programozási hibák*kal találkoznak. Természetesen remélhetőleg nem a már kész programokban, hanem azokban, amelyeket saját maguk készítenek, amíg az első, esetleg még a gép számára is érthetetlen változattól eljutnak a végleges, hibátlan megoldásig. Gyakorlati tapasztalat, hogy a programozók munkaidejüknek mintegy felét a programok helyességének ellenőrzésére (idegen szóval tesztelésére), a hibák megtalálására és kijavítására fordítják. A hibák megkeresésére általános érvényű szabályokat nemigen lehet adni, minden programozónak saját, vagy az okosabbaknak más kárán kell tanulnia. Ha valaki napokat töltött el egy ravasz programhiba megkeresésével, akkor biztos, hogy ugyanezt a hibát nem fogja mégegyszer elkövetni. Vagy ha mégis, akkor másodszorra sokkal hamarabb – esetleg már néhány óra alatt – rájön a hiba okára és többet már egészen biztosan nem fog nála előfordulni. De ha mégis, akkor már 10 perc alatt kijavítja és ezután már tényleg egészen bizonyosan . . . – ez majdnem olyan, mint egy végtelen ciklus, amelybe minden valamirevaló programozó többször is beleesett már – de előbb-utóbb valahogy csak kimászott belőle.

szintaktikai hibák

A programozási hibákat két nagy csoportra oszthatjuk. A *szintaktikai hibák* a programozási nyelv szabályainak megszegéséből származnak. Az ilyeneket a fordítóprogram vagy azonnal, az utasításnak a gépbe való beírásakor, vagy a fordítás befejezésekor kijelzi. (Egyes hibák, mint például egy elírt és ilyen módon nem definiált változó egy Cobol vagy PL/1 programban vagy egy hiányzó aritmetikai műveletjel az utasítás lefordításakor azonnal észlelhetők, de egy – sokak szerint eleve kiirtandó – GO TO utasítás, melynek ugrási címe az utasítás kiadásakor még nincsen definiálva, vagy egy *END* nélküli *BEGIN* csak a teljes program értelmezése után vehető észre.)*

Egy fordítóprogram használhatósága nagymértékben függ attól, milyen hibaüzeneteket ad ki, mert ezek segítik a programozót a hibák értelmezésében. Jobb fordítóprogramok megadják a hiba részletes leírását (sajnos általában angolul) és pontos helyét is az adott utasításban. Vannak azonban olyanok is, melyek csak egy hibaszámra hivatkoznak és a programozónak kell utánanéznie, milyen hibát takar ez a szám. E sorok írói is kénytelenek VolTak egy időben olyan géppel is dolgozni, melynek erre vonatkozó kézikönyvében a leggyakoribb hibaszám mellett csak ennyi állt: „szintaktikai hiba". A jó fordítóprogramok a szintaktikailag helyes, de logikailag minden valószínűség szerint helytelen

* Szinte minden programozási nyelvben megtaláljuk – esetleg más néven – a GO TO utasítást és a BEGIN – END jelölőspárt. Az előbbi azt jelenti, hogy (feltételtől függően, vagy attól függetlenül) a programot nem a soron következő, hanem a GO TO utasításban meghatározott helyen levő utasítással kell folytatni. A BEGIN és az END utasításokat foglalnak csoportba. Szerepüket a matematikából jól ismert kezdő- és végzárójelhez hasonlíthatjuk, azzal a különbséggel, hogy nem számokat, aritmetikai kifejezéseket, hanem programutasításokat vonnak egy csoportba.)

utasításokról is figyelmeztetést adnak (pl. egy 10 bájt hosszúságú adatot akarunk átvinni egy ennél rövidebb adatmezőbe, vagy egy feltétlen ugró utasítást közvetlenül követ egy olyan utasítás, amelyikre nem lehet máshonnan ráugrani. Ezeknél a példáknál a karakterek elveszhetnek, illetve az utasításra a program sohsem juthat el).

Elvileg vannak már olyan matematikai módszerek, amelyekkel be lehetne bizonyítani egyes programok logikai hibátlanságát, de ezek alkalmazása olyan bonyolult, hogy a gyakorlatban még nemigen használják fel.

szemantikai vagy logikai hibák

A *szemantikai* vagy *logikai hibák,* amelyek következtében a gép nem a programozó által kívánt műveleteket hajtja végre, általában csak úgy deríthetők fel, hogy a programot olyan gondosan kiválasztott tesztadatokkal futtatják le, melyekre az eredmény kézi számolás vagy más módszer alapján eleve ismert és a kapott eredményeket összehasonlítják.

A megfelelő *tesztadatok* kiválasztása igen nehéz feladat, mert elvileg az összes lehetséges adatkombináció típusnak legalább egyszer szerepelnie kellene bennük. Ez természetesen nem lehetséges a változatok nagy száma miatt. Ezért a gyakorlatban megelégszünk azzal, hogy ezeknek csak egy szűkített részére végezzük el az ellenőrzést. Mindenképpen tartalmazzák a tesztadatok a legvalószínűbb adatkombinációkat, a zérus, a megengedett határokkal egyező és azon kívüli értékeket, hogy valamelyik adatforrásból egyáltalán nem jön adat, illetve az utolsónak vélt adat után még egy jön, mi történik egy ciklus első és utolsó végrehajtásakor és ha azt egyáltalában nem kell végrehajtani. A következő fejezetben bemutatunk néhány jellegzetes programozási hibát, melybe kezdő programozók rendszeresen, gyakorlottak csupán időnként esnek bele. Természetesen tisztában kell lennünk azzal, hogy nagy programoknak ilyen tesztadatokkal való ellenőrzése nagyon hatékony lehet a programhibák jelenlétének kimutatására, de sohasem szolgálhat bizonyítékul a program hibátlanságára!

tesztadatok

A *tesztadatok* teljeskörűségének fontosságára nézzük a következő példát. Ha egy program a személyi szám második és harmadik számjegyéből állapítja meg a születési évet, és ebből az életkort, akkor ez nagy valószínűséggel csak Magyarországon használható. (A személyi szám második és harmadik jegye nálunk a születési év utolsó két számjegye.) Itt csaknem mindig helyes eredményt fog szolgáltatni. De annál az öreg bácsinál, aki 1897-ben született, azt fogja megállapítani, hogy az illető még csak −10 éves. (1997-et tekinti születési évnek.) Ezt a hibát a programozó, vagy a rendszer szervezője okozta, amikor nem építette bele a programba azt a logikai feltételt, hogy a múlt században születetteknél 3 vagy 4 a személyi szám első jegye. Ezért a születési év kiszámításánál ezt is figyelembe kell venni. Ha a próbaadatok között szerepelt múlt században született személy, akkor az ezekkel kapott eredmények figyelmes átvizsgálásával ez a hiba még a program üzembe helyezése előtt kiderül.

Lényegesen eredményesebben tesztelhetjük az olyan programokat, amelyeket kisebb, önállóan is ellenőrizhető részekre, szegmensekre bonthatunk. Az egyes kisebb szegmenseknél a lehetséges adatkombinációk száma jóval kevesebb, a program minden ágát le tudjuk vizsgálni. Ha a szegmensek külön-külön hibátlanok, akkor csak összekapcsolásukat kell ellenőriznünk, ami − a szegmensek számának korlátozott volta miatt − szintén megvalósítható teljeskörűen.

A fenti elv előnyeinek bemutatása végett nézzük a következő egyszerű példát: tekintsük az alábbi „programot":

$$X = (A+B)*(C-D)$$

Az adatokat (A,B,C,D) megadhatjuk bájtos, bináris, tömörített decimális, vagy lebegőpontos formában, és ezek bármilyen kombinációja is megengedett (lásd a 16. fejezetet!).

Tegyük fel, hogy a tényleges aritmetikai műveleteket a gép mindig lebegőpontos formában végzi el. Feladatunk annak ellenőrzése, hogy a különböző adatátkonvertáló műveletek vagy szubrutinok helyesek-e.

Ha a programot mint teljes egységet vizsgálnánk meg, úgy a különböző adattípusokból előállítható, a teljes ellenőrzéshez szükséges tesztadatkombinációk száma $4^4 = 256$ lenne. Ha a program szerkezete a tesztelhetőséget is figyelembe veszi, és az alábbi három önálló szegmentumra „bomlik",

$$X1 = A+B \qquad (1)$$
$$X2 = C-D \qquad (2)$$
$$X = X1 * X2 \qquad (3)$$

akkor mind (1), mind (2) $4^2 = 16$-féle adattípus kombinációt jelent. Mivel a számítások mind lebegőpontos aritmetikában történnek, (3) csupán egyfajta módon, lebegőpontos adatokkal valósulhat meg. Így ha a „programszegmentumokat" önállóan teszteljük le, és az így teljesen ellenőrzött szegmentumokat illesztjük össze, akkor a különböző esetek összes változata 16+16+1=33. Ez lényegesen kevesebb, mint a nem szegmentált program 256 variációja, így a tesztelés jóval kevesebb szellemi ráfordítást és gépidőt igényel.

Nagymértékben megkönnyíti a logikai hibák felderítését a programok futásának nyomon követése. Nagyszámítógépek erre közvetlen lehetőséget adnak akár úgy, hogy a végrehajtás folyamán az ellenőrizni kívánt, címkével megjelölt utasítások címkéjét — ha eljutott erre a helyre — a program automatikusan kiírja, akár úgy, hogy meghatározott helyeken a programozó által kért változók értékét, melyekből a további feldolgozás menete függ, kimutatja.

A professzionális személyi számítógépek megjelenésével a programtesztelés is párbeszédessé vált, a felhasználó a program futását a monitoron tudja követni. Ilyen környezetben méginkább megnőtt a futás közbeni hibafelderítő programok szerepe. Ilyen program segítségével sok minden megoldható: így például az éppen végrehajtott utasítás megjelenik a képernyőn, előre meghatározott helyeken a program megállítható, és a különböző változók értéke megkérdezhető. Tártartalmak felülírhatók, ha a továbbfuttatás ezt igényli. Mindezek a szolgáltatások nagymértében lerövidítik a hiba helyének behatárolásához szükséges időt.

Természetesen minél több programot készítettünk, annál nagyobb lesz a gyakorlatunk a programok tesztelésében, a hibák megkeresésében és kijavításában. És annál inkább magunkénak fogjuk érezni a számítástechnikai alapigazságot:

„Hibátlan program nincs! Csak olyan van,
amelyiknek a hibáit eddig még nem vették észre."

49.

BOCSÁSD MEG A VÉTKEINKET!
(Gyakori programozási hibák)

Tévedni emberi dolog, amint arról már könyvünk előszavából is értesülhetett az olvasó. Továbbá: csak az nem hibázik, aki nem dolgozik. Vagyis a programozási hibákból szeretnénk egy csokrot átnyújtani, azzal a szándékkal, hogy (még egy közmondást idézve) más kárán tanuljon az okos.

A bemutatásra kerülő hibákat nem mi találtuk ki, hanem több mint másfél évtized alatt tapasztaltuk, láttuk, néha talán magunk is elkövettük. Mivel e hibák egy része programnyelvtől függetlenül előfordul kezdő programozóknál, úgy tűnik, ezek gondolkodásunk és nem programozási ismereteink hiányosságaiból adódnak.

Nagyon gyakori a feladat, hogy összeg, átlag, vagy előfordulás (gyakoriság) kiszámításához adatokat kell összegezni. Ezt a legtöbb nyelv ciklussal oldja meg, amelyben minden menetben egy újabb érték adódik a korábbi részösszesenhez. Ha elfelejtjük a gyűjtőt a ciklus előtt nullázni, vagyis tartalmát nullával feltölteni, váratlan dolgok történhetnek. Vannak gépek, amelyeknek operációs rendszere az adatterületet üresen adja át a programnak. Vannak azonban olyan gépek, amelyek ezt nem teszik meg. Ilyenkor valamilyen „hulladék" bitsorozat van majdnem minden címen. A gyűjtést tehát a program úgy valósítja meg, hogy ehhez a szeméthez kezd hozzáadogatni. Ez a hiba akkor kellemetlen igazán, ha a program tesztelését és futtatását nem ugyanazon a gépen végezzük, mert ekkor (szerencsétlen esetben) az éles futáskor lép fel a hiba. Minden gyűjtőt nullázni kell tehát, mielőtt használatba vennénk.

Ha már a nullánál tartunk, gyakori hiba, hogy a program nincs felkészítve a nullával való osztás kivédésére. Ez elvileg minden olyan osztásnál előfordulhat, amelynek nevezőjében nem konstans érték van, hiszen lehet olyan szituáció, amikor a leendő osztó nulla értéket vesz fel. Célszerű tehát minden osztás előtt megvizsgálni, hogy az osztó nem nulla-e.

Majd minden programnak van adatbemenete, vagyis billentyűzetről vagy valamilyen más perifériáról adatigénye. Az a programozó, aki nem tanulta meg, hogy minden adat, amelyet egyszer a tárba juttatott, a program futása során ott marad, míg felül nem írjuk, vagy ki nem kapcsoljuk a gépet, előszeretettel olvastatja ugyanazt az adatot minden esetben, mielőtt használná azt, pl. egy művelet operandusaként. Ez nemcsak felesleges, hanem hiba is, hiszen egy szekvenciális állományból nem lehet kétszer (többször) egymásután ugyanazt az adatot elérni.

Az előbbi hiba egy újabbhoz vezethet. Ha egy állományból egy adatot vagy adategyüttest többször akarunk olvasni, ez azt jelenti, hogy az állomány hamarabb elfogy, mint kellene. Az állomány túlolvasása (vagyis a vége jel utáni olvasás) sok rendszernél olyan hibajelet generál, amelytől a program futása megszakad. Ez a hiba minden esetben előfordulhat, ha nem jól kezeljük a „vége az állománynak" információt.

Voltaire mondta, hogy „Istenem szabadíts meg a barátaimtól, az ellenségeimmel elbánok magam is". Ennek a mintájára született az a programozói fohász, amely így hangzik: „Istenem szabadíts meg az első és utolsó rekordtól, a többit én is fel tudom dolgozni." A baj leggyakrabban az első és az utolsó rekorddal (adategyüttessel) van, ha egy állo-

mányt kell feldolgozni. Tudniillik mindkettő speciális abból a szempontból, hogy előtte vagy utána nincs másik rekord. A programot tehát tesztelni kell arra az esetre, mintha csak egyetlen rekord lenne, valamint különös figyelmet kell fordítani, hogy mit csinál az utolsó rekorddal.

Ha szót fogadunk az előző tanácsnak, azonnal egy másik hibára is fel kell készülnünk. Tesztelni kell a programot arra az esetre, ha egyetlen rekord sincs! Az amatőr azt mondja, hogy ez az eset ki van zárva. A profi nemcsak azt tudja, hogy semmi sincs kizárva, hanem arra is tud példát, amikor egy állományban egyetlen rekord sincs. Ez fordulhat elő, ha az állományban egy időszak változásait rögzítő rekordok vannak, de az adott időszakban nem történt változás. Ezt az üzemeltető nem tudhatja, és elindítja az üres változásállománnyal a feldolgozást. A jó program ilyenkor kijelzi, hogy nem volt változás és befejeződik.

A sort még vég nélkül lehetne folytatni, de egyszer csak ki kell tennünk az állomány-vége jelet, egyszer mindent abba kell hagyni. Ezért befejezésül két klasszikus hibát mutatunk meg. (További „rossz" példákat találhat az olvasó az 52. fejezetben.) Az első példánk stílszerűen éppen az abbahagyással kapcsolatos. Nem is igazi programozó az, aki életében legalább egy *végtelen ciklust* ne írt volna. Ezt sokkal könnyebb megcsinálni, mint gondolnánk. Ha ugyanis sikerül egy feltételt úgy megfogalmazni, hogy az soha ne teljesüljön, és ettől a feltételtől tesszük függővé a ciklus befejezését, máris kész a végtelen ciklus. (Például egy függvényérték számításnál 0,2 lépésközzel növeljük a helyettesítési értéket a [0 ; 1] intervallumban. Ha a feltételünk x = 1, máris hibáztunk, mert a belső bináris ábrázolásban a 0,2 végtelen szakaszos kettedestört, így soha sem ad egzakt pontosságú 1-et ezek összege.) Ilyenkor nincs más, mint erőszakkal lelőni a programot. A nagyszámítógépeknél — helyesen megszervezett üzemeltetési rendszerben — éppen ezért minden programnak kötött futási időt adnak. Így az ottani végtelen ciklusok ezen idő túllépésekor megszakadnak.

végtelen ciklus

Másik példánk esetében egy állományt kell listázni (pl. dolgozók adatait, vagy leltáradatokat), de oly módon, hogy egy-egy lapra meghatározott számú tétel kerüljön, és minden lapon legyen fejléc, amely a címen kívül a fejrovatok magyarázó szövegét és lapszámot tartalmaz. Tíz kezdő programozó közül kilenc olyan megoldást készít, amely a legelső lapra nem csinálja meg a fejlécezést. S ezek között biztosan lesz olyan is, aki az utolsó utáni lapra (feleslegesen) egy plusz fejlécet nyomtat. (Összességében tehát minden rendben — mondja, ha elég nagyvonalú.)

De erre csak az alábbi, ismert mondással válaszolhatunk:

„Ez több mint bűn, ez hiba!"

50.

TÍZ PARANCS
(Operációs rendszerek gyakori parancsai)

Az egyes számítógép-típusok operációs rendszereikben is különbözők. Ezek a különbségek a felhasználó számára a lehetséges parancsok funkcióiban, formájában, hatásában jelentkeznek.

A nagyszámítógépek operációs rendszerei közül az egyik leggyakoribb rendszer, az IBM és a nagyobb ESZR gépeken használatos OS operációs rendszer három parancsát mutatjuk be, elsősorban azzal az igénnyel, hogy észrevehető legyen a különbség a többfelhasználós nagygép és az egyfelhasználós személyi számítógépek „lelkivilága" között.

JOB

A nagygépen fontos funkció a gépnek átadott munka azonosítása. Erre a *JOB* parancs használható. Ebben meg kell adni azt az azonosítót, amely a munkát végigkíséri a feldolgozás során. Minden e munkával kapcsolatos információ, amely a gépben megszületik, ezzel a névvel lesz azonosítva (hibaüzenetek, a felhasznált gépidő regisztrálása stb.). Ugyancsak a JOB parancsban kell közölni azt az elszámolási számot, amelynek alapján a felhasznált gépidőt elszámolják. (Bérmunkát végző számítógépeknél nagyon fontos.) A 49. fejezetben szó volt a végtelen ciklusról. Nos, az üzemeltető a JOB parancsban adhat meg – ennek elkerülésére – időkorlátot, amelynél több időt a központi vezérlőegység nem használhat fel a végrehajtás folyamán.

EXEC

A végrehajtásra kijelölt programot egy *EXEC* nevű parancs paramétereként kell megadni. Ilyen paranccsal aktivizálható pl. bármely fordítóprogram, szervizprogram, de ezzel indíthatja el a felhasználó valamely saját programjának a futását is.

DD

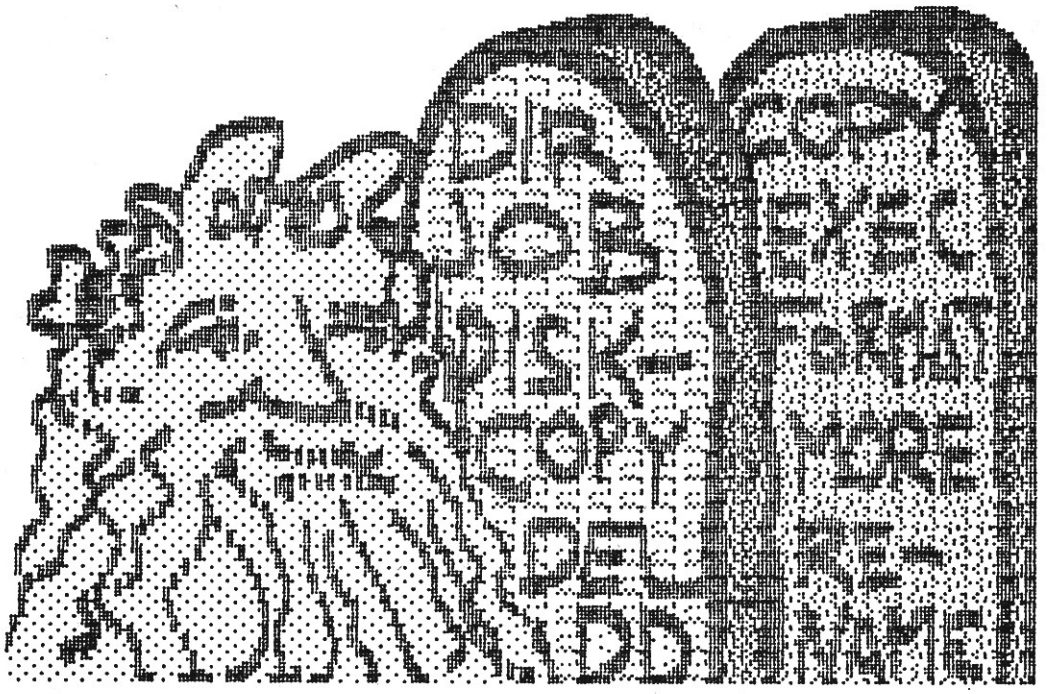
Minden futtatandó programnak van a környezettel adatkapcsolata. Minden állományt külön adatleíró, úgynevezett *DD* paranccsal kell leírni. Egy DD parancs tartalmazza az állomány fizikai helyének pontos meghatározását (pl. melyik lemezen van, mi a neve stb.) és belső szerkezetének leírását (pl. hány bájt hosszú rekordból áll, milyen a szervezettsége stb.).

Ilyen parancsokkal csak az találkozik, aki nagygépen old meg feladatokat. Könyvünk olvasóinak többsége azonban mikrogépen dolgozik. Az otthoni mikroszámítógépek közül például a Commodore 64 operációs rendszere viszonylag nehezen kezelhető. Ennek oka az, hogy az elsősorban Basic programozásra tervezett gépek az átlagos felhasználó számára szándékosan nem biztosítanak az operációs rendszerhez hozzáférést.

Egy 16 bites professzionális személyi számítógépnél viszont már nagyon sok szolgáltatást nyújt az operációs rendszer. E gépeken szinte egyeduralkodó az MS DOS operációs rendszer, amelyet néha PC DOS-nak neveznek a gyártók. Ismerkedjünk meg ennek az operációs rendszernek néhány fontos, és gyakran szükséges parancsával:

FORMAT

Amikor egy hajlékonylemezt használatba akarunk venni, szükséges, hogy a boltban vásárolt, teljesen üres lemezfelületet alkalmassá tegyük az információk fogadására. Ezt valósítja meg a lemezformázás, amelyet a *FORMAT* paranccsal (format = formázás) kezdeményezünk. (A most bemutatásra kerülő parancsok funkcióit nagygépeknél szervizprogramok valósítják meg.)



Nagyon fontos, hogy a különböző állományokról másolatokat lehessen készíteni. Ez nemcsak arra jó, hogy egy-egy játékprogramot lemásoljunk (mert ezek általában védve vannak másolás ellen), hanem arra, hogy saját programjainkról készíthessünk másolatot, hiszen bármelyik lemezünk bármikor meghibásodhat, s ilyenkor esetleg minden rajta levő információ elvész. (Murphy szellemében: mindig az a lemez romlik el, amelyikről nincs másolatunk.) Állomány másolása a *COPY* paranccsal (copy = másol, kopíroz) történik. Meg kell adni a másolandó állomány nevét és a másolat nevét. (Ha a másolatot másik lemezre kérjük, a két állománynak lehet azonos neve.) A parancsnál nagy segítség lehet a * jelnek a használata. (Az MS DOS-ban egy állomány neve az alábbi formájú: meghajtónév:név.kiterjesztés. Például a c:minta.for név azt jelenti, hogy a kérdéses állomány a C jelű meghajtón levő lemezen van. MINTA a neve és FOR kiterjesztésű, ami arra utal, hogy Fortran programról van szó.) Ha a COPY parancsban *-ot adunk meg, ez helyettesíti az állomány nevét vagy a kiterjesztést.

COPY

Például a "COPY a:*.for b:" parancs jelentése: az A meghajtóban levő lemezről a B meghajtóban levő lemezre másolandó minden FOR kiterjesztésű állomány. Ugyanígy a "COPY a: orsi.* b:" jelentése: írjon át a B-n levő lemezre az A-ról minden ORSI nevű állományt, függetlenül attól, hogy milyen a kiterjesztésük. A "COPY *.*" azt jelenti, hogy minden állományt másolni kell. A * tehát a kártyából ismert Jolly Joker szerepét tölti be.

DISKCOPY

A teljes lemez a *DISKCOPY* paranccsal is lemásolható (disk = lemez; diskcopy = teljes lemezt lemásolni). De álljon itt egy jótanács parancs képében: teljes lemezmásolatot célszerűbb a COPY *. * formában kérni. A DISKCOPY parancs ugyanis a lemezről formailag is pontos másolatot készít, s ha másolandó lemezen pl. hibás sáv volt, ez a sáv a másolaton is üres marad.

DEL

Amikor egy állományra nincs már szükség, le kell törölni. Ezt a *DEL* paranccsal tehetjük meg. Az MS DOS egyes változatai még egy megerősítést kérnek a parancs kiadása után (megkérdezik, hogy tényleg töröljék-e a megadott állományt). Jegyezzük meg, hogy az állományokról rendszerint csak a törlés után derül ki, hogy mégis kellenének valamire.

Gyakori igény, hogy tudni szeretnénk, mi is van az éppen használt lemezen. Ezt a funkciót, vagyis a lemez tartalomjegyzékének kiírását a *DIR* paranccsal lehet megoldani (DIRectory = tartalomjegyzék). A tartalomjegyzék a képernyőn jelenik meg.

Minden állományról a következő információk jelennek meg:

DIR

- az állomány neve,
- az állomány mérete (bájtban),
- az utolsó változtatás dátuma,
- az utolsó változtatás időpontja.

Ha 23-nál több állományunk van a lemezen, akkor az állománynevek nem férnek el mind a képernyőn. Ilyenkor a képernyő tartalma „elindul" fölfelé. Az újabb állománynév a képernyő alján jelenik meg, a legtetején levő pedig eltűnik. Az összes többi egy sorral feljebb kerül. Mindez olyan gyorsan történik, hogy nem tudjuk a kifutó információt elolvasni, ami megakadályozható, ha a DIR parancsot másképp adjuk ki.

A DIR/P hatására az operációs rendszer leállítja a listázást, ha a képernyő betelt. A megjelenítés folytatódik bármely billentyű lenyomása után.

Ha már elfelejtettük, hogy mi is van egy adott nevű állományban, megnézhetjük az állomány tartalmát a *MORE <állománynév* formában kiadott paranccsal.

MORE

A MORE parancs a megadott állomány tartalmát szakaszosan jeleníti meg. Ez a program a megadott állományból beolvas egy képernyőnyi információt és azt megjeleníti. Ha az állománynak nem ért a végére, a képernyő legalsó sorában a

——More——

üzenet jelenik meg. Tetszőleges billentyű leütésére a következő képernyőnyi információ kerül a képernyőre. (Ez a parancs tehát programot indít el.)

Néha szükséges egy meglevő állomány nevének a megváltoztatása. Ez a *REN* (esetleg *RENAME* = újra elnevez) paranccsal lehetséges. Meg kell adni a régi és az új nevet.

REN
RENAME

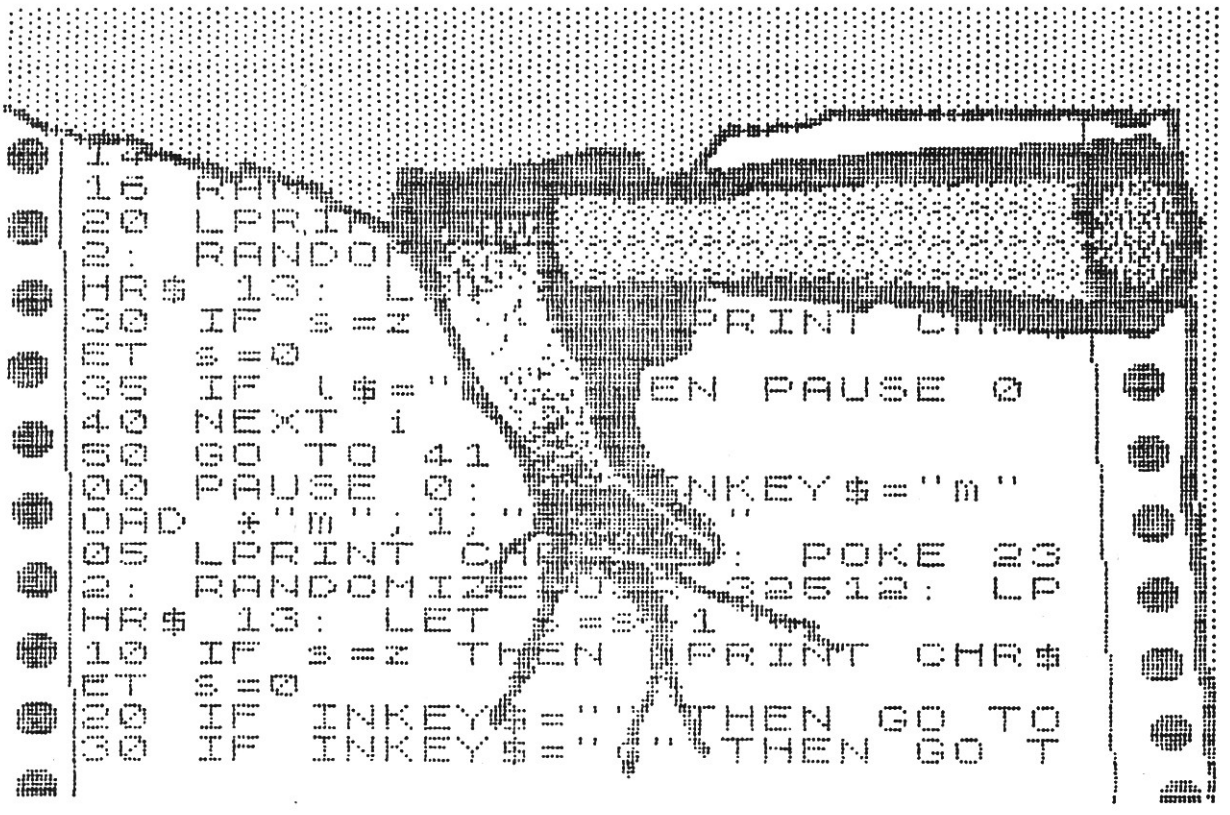
A MORE példáján láttuk, hogy vannak parancsok, amelyek mögött egy azonos nevű program van. Ezek a 31. fejezetből ismert külső parancsok. Bármely végrehajtható programnak elég megadni a nevét, a rendszer azt a programot betölti és elkezdi végrehajtani. (Így indítható pl. a kívánt fordítóprogram, vagy a saját futtatható programunk.) Aki volt katona, tudja, hogy a parancsot végre kell hajtani, mert ha mást teszünk vele, szaporodik. A számítógép esetében viszont így lehet tíz parancsból akár százat is csinálni.

51.

ÍGY PROGRAMOZTOK TI (ÉS ŐK IS)
(Számítástechnika alulnézetben, a leggörbébb szoftveres szemmel nézve)

A számítástechnikai rendszerek, programok kifejlesztéséhez szükséges idő meghatározására jól bevált az alábbi módszer. A rendelkezésre álló összes adat alapján pontosan kiszámoljuk a munka elvégzéséhez még a legrosszabb feltételek esetén is elegendő időt. Ezt megszorozzuk kettővel és a mértékegységet egy nagyságrenddel megnöveljük. Így egy maximálisan egy napig tartó feladatot két hetes határidőre vállalunk. És ha mégsem leszünk kész vele, ígérjük meg másnapra. Mert mi Murphyval ellentétben optimisták vagyunk. Hátha nem lesz másnap!

Az idő furcsa szerepet játszik a programozásban. Relatív! A főnökök szerint, ha *n* elég nagy, akkor *n+1* egyszerű program elkészítéséhez ugyanannyi időre van szükség, mint n program megírásához. A programozók szerint kétszer annyira.



Programhiba

Általános tapasztalat, hogyha egy határidőcsúszásban levő szoftver fejlesztésére több embert állítanak rá, akkor a határidő még jobban kitolódik. Ennek magyarázata az anyatermészetben rejlik. Egy férfi, ha gyereket akar, megtermékenyít egy nőt és kilenc hónap múlva lesz egy gyereke. De ha sürgősen akar egy csemetét, hiába termékenyít meg egyszerre kilenc nőt, nem lesz egy hónap múlva gyereke. Legföljebb 9 hónap múlva 9. Persze örökbe fogadni is lehet. Ezt hívják a számítástechnikában programtermék-vásárlásnak.

Ha pedig a késedelem behozása érdekében a tehetetlen vezető helyett újat neveznek ki a projekt élére, igazolódik az alábbi törvény is:

Az impotencia öröklődik.

Jó programozók — Murphy alaptörvényének továbbfejlesztése következtében — természetesen tisztában vannak az alábbi alapvető igazságokkal:

☆ Nincs hibátlanul futó program. De ha lenne, akkor arra már senkinek sincs szüksége.

☆ Ha mégis kellene valakinek, akkor biztos meg kell változtatni.

☆ Ha egy újonnan elkészült programra senkinek sincs szüksége, akkor azt a programot precízen le kell dokumentálni.

☆ Minden program oly mértékben bonyolódik a fejlesztés folyamán, amíg áttekinthetetlenné nem válik azok számára, akiknek azt el kell készíteni.

☆ Ha egy hibát felfedezünk és kijavítunk egy programban, akkor rövidesen kiderül, hogy az eredeti változat volt a helyes.

☆ Miután kiderült, hogy a javítás volt a hibás, képtelenek leszünk a programot az eredetileg is helyes állapotába visszaállítani.

Még az a szerencse, hogy talán jó programozók sincsenek és így a fenti „igazságok" is csak félig fedik a valóságot. Ennek görbe tükrében az utolsó két „tapasztalatot" még több gyakorlat alapján így vonhatjuk össze:

☆ Ha egy hibát felfedezünk és kijavítunk egy programban, majd kiderül, hogy a javítás volt a hibás, hosszú ideig képtelenek leszünk a programot az eredeti állapotába visszaállítani. Amikor az végül sikerül, akkor azonnal nyilvánvalóvá válik, hogy a visszaállított, eredeti változat *sem* volt helyes.

Erre a törvényre hivatkozva, ha könyvünkben véletlenül egy-két hibát, pontatlanságot fedezne fel a tisztelt *lovasó,* akkor annak nem a szerzők és a lektorok felületessége az oka. Ezen törvények következményeit ismerve tudjuk, hogy semmilyen hibát sem szabad kijavítani.

52.

AZ AMATŐR PROFI ÉS A PROFI AMATŐR
(Szakma-e a számítástechnika?)

Társasági vitákat, baráti és folyosói beszélgetéseket hallgatva gyakran az az érzésünk, hogy Magyarországon mindenki ért a focihoz (még ha ezt a Mundial '86 „eredményei" cáfolták is) és a . . .-hoz. Ez a három pont itt és most nem egy nyomdafestéket nem tűrő szóval helyettesítendő, hanem a helyén voltaképpen bármilyen szakmának, foglalkozásnak, tevékenységnek a megnevezése állhat, a társaságtól függően: a gyermeknevelés, az irodalom, a tévéműsor megszerkesztése, az építészet vagy a közgazdaságtan stb. Újabban, a személyi számítógépek elterjedésével pedig sokan úgy vélik, hogy hamarosan mindenki, de legalábbis az iskolából most kikerülő nemzedék nemcsak használni tudja majd a számítógépet, hanem szakember is lesz ezen a területen. S ahogy az ember elnézi, hogy a mai tizenévesek milyen magabiztossággal nyomkodják a számítógép billentyűit, úgy érezheti, hogy ez az óhaj valóban beteljesedik.

Tény, hogy a fiatalok nagy része bámulatos gyorsasággal megbarátkozott a számítógéppel, a különféle számítógépes játékokkal. Emlékszem, hogy amikor öt évvel ezelőtt megvásároltunk egy ilyen játékot, családunknak felnőtt — a számítástechnikával hivatásszerűen foglalkozó, abból élő — tagjai a leírás gondos áttanulmányozása után is legföljebb 50—60 pontot tudtak elérni vele egyhetes gyakorlással. Ugyanakkor 19 éves fiunk és 11 éves lányunk — anélkül, hogy egyetlen szót is értettek volna a német nyelvű kezelési utasításból, csupán bennünket figyelve (gúnyos mosollyal) és az utasítás ábráiból, meg főként a képzelőerejüknek, a merészségüknek és ösztönösen jó hozzáállásuknak köszönhetően — már az első nap vidáman túlszárnyalták a mi eredményeinket, s néhány nap múltán már 1000 pont körül tartottak.

Hasonló, sőt, még meglepőbb története szinte mindenkinek van. De vajon valóban azt jelenti-e ez, hogy a gyermekeink értenek a számítógéphez, hogy ők már számítógépes szakemberek?

Amikor egy bonyolult eszköz vagy berendezés közhasználatba kerül — ilyen volt például egy évszázaddal ezelőtt a gépkocsi, néhány évtizede pedig a számítógép —, kezelése többnyire komoly szakértelmet kíván meg mindenkitől. A századfordulón akinek volt pénze autót tartani, annak sofőrre is tellett, s általában szüksége is volt rá: ő maga nem akarta, nem tudta vagy nem merte irányítani az óránként több mint 30 kilométeres sebességgel „száguldó" masinát. És ma? Egy rövid tanfolyamon szinte bárki megtanulhat autót vezetni, holott a forgalom és a sebesség azóta többszörösére növekedett! Ám a gépjárművezetés azóta is szakma maradt. Hiszen egy autóbuszt vagy akár egy taxit egész napon át vezetni a városi forgalomban nem ugyanaz, mint hét végén kiautókázni a telekre.

A számítástechnika hős- és őskorában, amikor még szinte minden gép más és más volt, a feladatokat a gépek tervezői, általában kiváló szakképesítésű matematikusok oldották meg. Aztán ahogyan nőtt a számítógépek és a feladatok száma, megjelent a színen az a szakember, aki a munkát gépre szervezte, megírta a feladat megoldásához szükséges programot. Ahhoz persze, hogy a rutinfeladatokat ne csak „zsenik", hanem az oktatási intézményekből ezrével kikerülő „jó iparosok" is el tudják végezni, meg kellett változnia

magának a számítógépnek is. A munkamegosztás a hatvanas években, a számítógép harmadik nemzedékének az elterjedésével vált teljessé. Ekkorra sikerült a számítógépek építését, karbantartását, üzemeltetését, programozását és a gépre viendő feladatok elemzését oly mértékben szabványosítani, hogy kialakulhattak az önálló számítástechnikai munkakörök és szakmák (lásd a 7. fejezetet!). Ma bár ugyanazon csapatnak a tagjai, mégis bizonyos mértékig egymástól függetlenül dolgozik a rendszerszervező, a programozó, a számítógép-tervező, a -fejlesztő, a -karbantartó, az üzemeltető és az adatrögzítő stb., mert mindegyik részlegnek a feladata jól és magas fokon csak sajátos ismereteknek a birtokában és nagy gyakorlattal végezhető el.

Mint ahogyan az autók tömeggyártása nyomán — ha nem élesen, de kettévált az autót használók és az autós szakemberek köre, úgy a mikroszámítógépek révén a számítógépes szakemberek mellett megjelent és egyre rohamosabban növekszik a számítógépet értelmesen felhasználóknak a tábora is.

Néhány évvel ezelőtt a számítógépterem zárt terület volt, oda csak a kiválasztottak, a számítástechnikai szakemberek közül is csak kevesen juthattak be: a drága berendezéseket még a lehellettől is féltették. Olvasóink zöme is csak képzeletben, könyvünk 53. fejezetének segítségével pillanthat be egy „igazi" nagygéppel rendelkező számítóközpont géptermébe, hogy közvetlen közelről „láthassa", mi minden történik ott. A személyi számítógépek térhódításával azonban ma már nálunk is csaknem mindenki, aki akar, hozzáférhet valamilyen géphez. Aki pedig géphez jut, az csinál is vele valamit: játszik vagy programot ír. Ettől azonban — még ha a program sikerül is — még nem válik szakemberré, valójában nem tud még többet, mint jól-rosszul használni egy bonyolult berendezést.

Nem állítjuk, hogy az amatőr teljesítménye mindig gyengébb, mint a profié („hivatalosan" az olimpikonok még mindig amatőrök, míg az NB II-es labdarúgóink — státuszuk szerint — már profik). Mégis az igazi szakemberek többsége rutinszerűen, akaratlanul is jól dolgozik, míg az önképzőkörben nevelkedett „műkedvelők" programjáról sokkal gyakrabban derül ki, hogy az amatőrmunka volt a szónak a szokásos értelmében. (Ez, persze, nem jelenti azt, hogy a szakemberek csak kiváló, hibátlan terméket adnak ki a kezükből!)

Mert milyen is általában az „amatőrprogram"? Rendszerint néhány tucat vagy néhány száz adattal gyorsan és hibátlanul működik, ám rögtön használhatatlanná válik, mihelyt nagyon kevés vagy nagyon sok adattal kellene dolgoznia. Ha kevés az adat, előfordulhat, hogy valamelyik adatcsoportba egyetlen egy elem se kerül bele (az adatállomány üres marad), s ezért a programutasítás szerint például nullával kellene osztani. Ha pedig sok az adat, az adatoknak az összege, szorzata stb. nem fér el a tárban, a számukra fenntartott helyen. Ha a program készítésekor nem gondolunk ezekre az esetekre (ha ilyen szélsőséges adatokkal nem vizsgáljuk meg, hogy miként működik a rendszer), a program vagy leáll (s ennek okát előbb-utóbb megtaláljuk), vagy végrehajtódik, és hibás eredményeket szolgáltat. Ez utóbbinál óriási kár származhat. Jellemző erre, hogy a szoftverek eladásakor a szerződésben mindig van egy apróbetűs bekezdés: ha a program hibás, azt készítői vagy forgalmazói díjmentesen kijavítják, újra való lefuttatásának költségeit megtérítik, de a téves eredmények felhasználásából eredő károkért semmiféle felelősséget sem vállalnak. Ha egy hibás programmal tervezett híd netán összeomlik, a felelősség ezért — jogilag — továbbra is a tervezőé és nem a program készítőjéé vagy forgalmazójáé.

Gyakorta adódhat hiba abból, amikor a gépnek néhány, jól meghatározott lehetőség közül kell egyetlen egyet kiválasztania. Nézzünk erre példát: csoportosítsa a gép az újszülötteket nemük szerint. Ha az újszülött nem FIÚ, akkor biztosan LÁNY. De mi van akkor, ha az adatbevitelkor melléütünk, ha az F helyett mondjuk G-t billentyűzünk be

a gépbe? Ha a program csupán a FIÚ jelzésű újszülötteket minősíti fiúnak, az összes többit pedig lánynak, akkor a fiúból itt menthetetlenül lány lesz. Az amatőr programozó gyakran elkövet ilyen hibát. És a profi? A tapasztalat szerint néha ő is, bár azért nem ilyen nyilvánvalóan egyszerűt. Miután gyakran az amatőrökből lesznek a legkiválóbb profik – minden nagy férfiú is volt egyszer kisfiú –, legjobb azt mondani, hogy aki ilyen hibát követ el, az egyszerűen rossz programozó.

Az amatőr bízik abban, hogy az adatokat még ha azok hosszú számok is, pontosan rögzítik, pontosan billentyűzik be a gépbe. A szakember már tudja, hogy mindenki hibázhat, ezért ha csak lehetséges, ellenőrző adatokat is beleépít a programba. Különösen gyakori ez, amikor az adatok csak számjegyekből állnak. Ilyenkor minden adatban egy (vagy két) számjegy csupán arra szolgál, hogy a többit ellenőrizni tudjuk. Így a gép, ha egy számban egy számjegyet a bevitel során hibásan adunk meg (például a billentyűzeten félreütünk), jelzi a hibát, s több számjegy félreütését is nagy valószínűséggel kijelzi. Ilyen *ellenőrző szám* van beépítve például a 11 jegyű *személyi számokba* is.

ellenőrző szám

Hogy ez valóban így van, ellenőrizzük saját és barátaink, rokonaink személyi számát! (Hogy ezzel egyben egyesek féltve őrzött életkorát megtudhatjuk, az már csak tiszta haszon – vagy keserű kiábrándulás.)

A személyi szám első számjegyét 1-gyel, a másodikat 2-vel, a harmadikat 3-mal, és így tovább, a tizedik jegyet tízzel szorozzuk meg. Adjuk össze az így kapott szorzatokat és vonjuk le belőle az utolsó számjegyet: eredményül 11-gyel osztható számot kell kapnunk. Például, e könyv egyik szerzőjének személyi számán, ami 13610160083 ellenőrizhetjük: $1\times1+3\times2+6\times3+1\times4+0\times5+1\times6+6\times7+0\times8+0\times9+8\times10-3=154$ és $154=14\times11$. Ha például az egymást követő 3-at és 6-ot felcseréljük, vagy a 8 helyett 9-et írunk, a végeredmény 157 illetve 164 lesz. Mivel egyik sem osztható 11-gyel, a hibát észrevesszük.

Matematikailag könnyen bebizonyítható, de ha nem hisszük el, akkor akárhány példán kipróbálhatjuk, hogy minden olyan hibát, amely egy számjegy téves megadásából adódik vagy bármely két számjegy felcserélésével követünk el (adatbevitelnél ez a két leggyakoribb hiba), ezzel az ellenőrzéssel *mindig* ki tudunk mutatni. Sőt, még két számjegy egymástól függetlenül történő téves megadását is csaknem mindig észrevesszük.

A házi számítógép legföljebb néhány száz adattal dolgozik, ezért majdhogynem mindegy, hogy az adatok miként helyezkednek el a tárolóban, a számolóművet miképpen fér hozzájuk. Mihelyt azonban nagy adattömeggel kell számolni, az adatok szervezése már egyáltalán nem hanyagolható el, mivel tárolásukhoz külső egységet is, általában mágneslemezt (vagy mágneslemezeket) is igénybe kell vennünk. Ha ezen az adatokat ötletszerűen helyezzük el, kikeresésük, rendezésük akár órákba is beletelhet, főként akkor, ha kicsi a gépünk. A hivatásos programozó bizonyos fogásokat alkalmazva ezt az időt számottevően – a tizedére, századára – csökkentheti.

A hivatásos és az alkalmi programozó munkája között a legnagyobb különbség a biztonságban van. Mihelyt többen is hozzáférhetnek egy adatállományhoz, a programba be kell építeni egy olyan ellenőrző rendszert is, amely szavatolja, hogy a programot csak az futtathassa, akinek ehhez joga van. Továbbá dokumentálni kell, hogy ki, mikor és milyen változtatásokat hajtott végre az adatokon. Arról is gondoskodni kell, hogy semmilyen hiba esetén se semmisüljön meg a program, illetőleg az adatállomány, nehogy hetek, hónapok, esetleg évek munkája menjen veszendőbe egy váratlan áramkimaradás miatt. Azoknál a rendszereknél pedig, amelyekben gyakran változtatják az adatokat, létkérdés, hogy a korábbi állapotot visszaállíthassák. Csak így kerülhető el, hogy akkor se vonják

le kétszer az OTP tartozást vagy a gyerektartási díjat, ha egy hiba miatt a programot újra kell indítani.

Egy programozási nyelv szabályait elsajátítva készíthetünk ügyes programokat, de ezért még ne tartsuk magunkat számítástechnikusnak! S még akkor sem, ha a technikus kevesebbnek tetszik, mint például a mérnök. A programozás is szakma, és ezt nem könnyű dolog jól elsajátítani. Ettől függetlenül, persze, mindenki nyugodtan írhat programokat. És ez a foglalatosság sokszor a szellemi munka örömén kívül még hasznos is lehet, miként általában a barkácsolás is az. (Milyen jó és mennyivel gyorsabb és olcsóbb is, ha magunk javítjuk meg a csöpögő vízcsapot, ha egy nő át tudja alakítani a tavalyi ruháit, sőt újakat is tud varrni magának és a gyermekeinek!) s örülhetünk annak, ha gyermekeink játszanak a számítógéppel, ha saját programjaikat futtatják. Mert ezt látva bízhatunk benne, hogy a felnövekvő nemzedék valóban együtt él majd a számítástechnikával, hogy azt értelmesen hasznosítja mindennapi munkájában. Az erre való felkészítésben nagy szerepe van az iskolai számítástechnikai oktatásnak, a különféle számítástechnikai kluboknak és táboroknak, s az otthoni személyi számítógépeknek is. De azért ne értékeljük túl az ezeken a fórumokon szerezhető ismereteket. Mert más dolog a számítástechnikát használni, alkalmazni és megint más azt művelni. Egy példával élve: még nem televíziós szakember az, aki be tudja kapcsolni a tévékészüléket, és nem lesz tv-kritikus, ha bírálja a műsort.

Mint ahogy önmagában attól nem válnak valakik íróvá, számítástechnikusokká, vagy egyidejűleg mindkettővé, mert megírtak egy számítástechnikával foglalkozó könyvet.

53.

BELÉPÉS CSAK KÜLÖN ENGEDÉLLYEL
(Látogatás egy számítóközpontban)

Nem volt könnyű helyzetben az, aki néhány évvel ezelőtt úgy határozott, hogy működés közben közelebbről megnéz egy valódi számítógépet. Ha az illetőnek nem volt az ismeretségi körében számítástechnikai szakembere, talán ez a vágya nem is teljesülhetett. Mindez ma már furcsán hangzik, hisz egyre több munkahelyen (és családban) található valamilyen számítógép. De egy számítóközpont megtekintése ma sem egyszerű dolog.

Csatlakozzunk (legalább gondolatban) egy kíváncsi csoporthoz, amelynek megadatott, hogy belépjen az oroszlánbarlangba, és látogassunk meg egy számítóközpontot. Kísérőnk, a központ vezetője, aki először a feladataikról tájékoztat bennünket:

— Oktatási számítóközpont vagyunk, ezért elsődleges feladataink közé tartozik a hozzánk küldött diákok számítástechnikai és szakmai tanulmányaihoz a számítógépes háttér biztosítása. Ezeknek a diákoknak a kiszolgálása terminálokon keresztül, többfelhasználós időosztásos rendszerben történik.

A diákokkal való foglalkozás mellett más feladataink is vannak — folytatja kísérőnk —, így mindenekelőtt az oktatás szervezésével és adminisztrációjával kapcsolatos rendszerek üzemeltetése, illetve újabbak létrehozása és programozása. Feladatunknak tekintjük továbbá egyes kutatások számítógéppel való támogatását.

Induljunk! — hallatszik vezetőnk hangja, és a zsebéből elővett műanyag lapocska segítségével nyitja a bejárati ajtó mágneses zárját. Néhány diák jön velünk szembe, most fejezték be (vagy csak abbahagyták) munkájukat. Elsőnek az „adatrögzítők" feliratú ajtón lépünk be. A tágas, világos szobában két lyukkártyalyukasztó gépet és egy négymunkahelyes mágnesszalagos adatrögzítő berendezést találunk. Az egyik lyukkártyalyukasztónál piros köpenyes nő ül. Kódlapra írt adatot rögzít lyukkártyára. A billentyűzetet szinte nem is nézi. Nagyon gyorsan dolgozik. A másik piros köpenyes lány egy képernyő előtt ül, és a billentyűzeten keresztül visz fel adatokat. Egy nyilvántartáshoz rögzíti a módosító rekordokat. A szoba közepén kis asztal körül kényelmes karosszékeket látunk.

Itt lehet pihenni — mondja vezetőnk —, tudniillik óránként szünetet kell tartani, mert sem derékkal, sem szemmel nem lehet sokáig bírni a rögzítést. Fárasztó ez a munka.

Tovább megyünk. Pár négyzetméteres helyiségbe lépünk be, ahonnan minden irányba ajtók nyílnak. Az egyiken tábla, amelyre ahelyett hogy ez lenne felírva: „Ki itt belépsz, hagyj fel minden reménnyel!", ez áll:

„Gépterem. Idegeneknek belépni tilos!"

Kísérőnk az ezzel szemben levő ajtón nyit be. — A diszpécserek — mutat maga elé — készítik elő a számítógépes feldolgozásokat, és tőlük veszik át a felhasználók az eredménytablókat. (Aki olvasta könyvünk 27. fejezetét, tudja hogy őnekik kell a csokoládét hozni.)

A szobában világossárga köpenyes fiatalember éppen a gépteremből kihozott listákat szedi szét. Munkáját megkönnyíti, hogy a kezdőlapon nagy betűkkel van felírva a program azonosítója.

– Itt vannak a terminálról indított programok nyomtatott eredményei is – mondja a fiatalember, és egy köteg leporellóval elindul a folyosóval határos fal felé. Beépített fehér szekrényhez lép. Kinyitja az egyik ajtót és a polcra beteszi a leporellókat. Ekkor látjuk, hogy a szekrény hátlapja hiányzik. A folyosóról nézve ez nem is szekrény, hanem egy beépített polcrendszer.

Visszatérünk a szűk előszobába, majd elindulunk a gépterem felé. – Ez a légzsilip – mondja kísérőnk –, azért van rá szükség, mert a gépterem klimatizált és pormentesített s ezért nem nyílhat közvetlenül a folyosóra. A gépterembe nyíló vasajtó egyébként tűzszakasz-határoló is. Itt pedig – mutat az egyik ajtóra – az operátori pihenő található, de most éppen üres.

Belépünk a gépterembe. Hűvös, száraz levegő csap meg. Kellemes világítás, tompa, de nem bántó zaj. „Ez lenne tehát az a hely, ahova egyszerű halandó soha nem juthat be?"

A gépterem rendjét írásban szabályozták és ez jól látható helyen van elhelyezve a falon. Itt találjuk a tilalmakat és a kötelezettségeket felsorolva (pl.: tilos a dohányzás, a gépteremben legalább két főnek kell tartózkodnia stb.).

Emelkedő rámpán teszünk néhány métert. A gépterem padlószintje legalább negyven centivel magasabban van, mint a légzsilipé, ahonnan beléptünk. „Ez az álpadló, alatta vannak a különböző egységek számára áramot biztosító, valamint az adatátvitelt szolgáló kábelek, s ez alá fújja be a klímagép a friss levegőt."

A mennyezetben süllyesztett lámpák, levegőelszívók és füstjelzők vannak. A gépteremben éppen nagy a nyüzsgés. A másik diszpécser egy kis szállítókocsit gördít az operátorokhoz. A kocsin mágneslemezek, kevés lyukkártya.

– Délután esedékes a havi feldolgozások egyike – magyarázza kísérőnk, amikor észreveszi, hogy a kocsit figyeljük. A diszpécser most adja át az operátornak a futtatáshoz szükséges mágneses adathordozókat és a feldolgozáshoz szükséges munkavezérlő kártyákat. Az operátor a futtatási lapot tanulmányozza, és ellenőrzi, hogy a szalag- és lemeznevek megegyeznek-e a lapon levőkkel. A másik operátor két barna köpenyes férfival beszélget.

– Főnök, a hármas lemezegység megint nyavalyog! – szól az egyik barna köpenyes, a hardvermérnök. Az egyik lemezmeghajtóról leszerelték a borítólemezeket, a teteje is nyitva van, szemlátomást „nyavalyog". Hirtelen erős zaj hallatszik. Egy sornyomtató meglehetősen nagy zajjal nyomtatni kezdett. A hardveresekkel beszélgető operátor odalép a nyomtatóhoz, ellenőrzi, hogy a papír jól van-e beállítva, majd figyeli a kijövő leporellókat. A négy mágnesszalagegység közül kettő van bekapcsolva, az egyiken a szalag szép tempósan halad előre. Időnként lelassul, majd ismét felgyorsul.

Az operátor egy képernyővel ellátott billentyűzeten ír valamit. Ez a konzol, amely biztosítja a kapcsolatot a számítógép operációs rendszere és a számítógépet kezelő ember között. Az operációs rendszer a konzolra küldi az üzeneteit, és az operátor ugyancsak ezen küldhet parancsokat az operációs rendszernek.

– Ez a központi egység, ugye? – mutat az egyik látogató a szemben levő, jelzőlámpácskáit villogtató fémszekrényre.

– Igen, ez a központi egység, és itt van a 2 Mbájtos központi tár is.

számítóközpontok tűzvédelme

Az egyik lány, akinek az apja tűzoltó, az oltókészülékek iránt érdeklődik. Édesapjától hallotta, hogy néhány éve egy számítóközpontban tűz ütött ki, és az oltáshoz porraloltót használtak. A por több kárt tett a gépben, mint a tűz.

– *A számítóközpontok tűzvédelme* nemcsak azért fontos, mert több millió forintos értéket képviselnek ezek a gépek, hanem azért is, mert a számítógépekben őrzött adatok,

információk is nagy értéket jelenthetnek. A számítóközpontokban speciális gázzal, halonnal töltött tűzoltókészülékek találhatók. Nézzétek, ott a bejárat mellett és a gépterem közepén vannak elhelyezve az oltókészülékek. Az operátorok megfelelő kiképzésben részesültek. A legfontosabb persze itt is a megelőzés. Indulunk kifelé.

— Nézzük meg a terminálokat is — invitál vezetőnk és indul is előre, hogy mutassa az utat. A gépteremmel szemben, a folyosó másik oldalán található a kiírás: „Terminálok". Gépterem nagyságú helyiségbe nyitunk be. A teremben egymás mellett ötösével terminálok sorakoznak. Húsz egyforma terminál s hátrább, a terem túlsó végében más típusú berendezések. Majd mindegyiknél ül valaki.

— A húsz terminálon kívül tíz professzionális személyi számítógép is a diákok rendelkezésére áll — kapjuk a magyarázatot halkan, hogy ne zavarjuk a munkát.

Megállunk az egyik hallgató háta mögött. Éppen programot ír, vagy ahogy itt mondják, editál. Eközben gondolkodik is — meditál. A készülő program utasításait egymás után írja fel a képernyőre. Ezzel egyidejűleg a bevitt sorok rögzítődnek egy mágneslemez munkaterületén. Majd ha kész lesz, az egészet egyszerre bemásolhatja a saját programkönyvtárába.

Tovább lépünk. A másik hallgató a már elkészült programját javítja. Egy rosszul begépelt változó nevét korrigálja, majd két új utasítást illeszt a programba. — Talán most jó lesz! — sóhajt és az egész programot visszaíratja a könyvtárába.

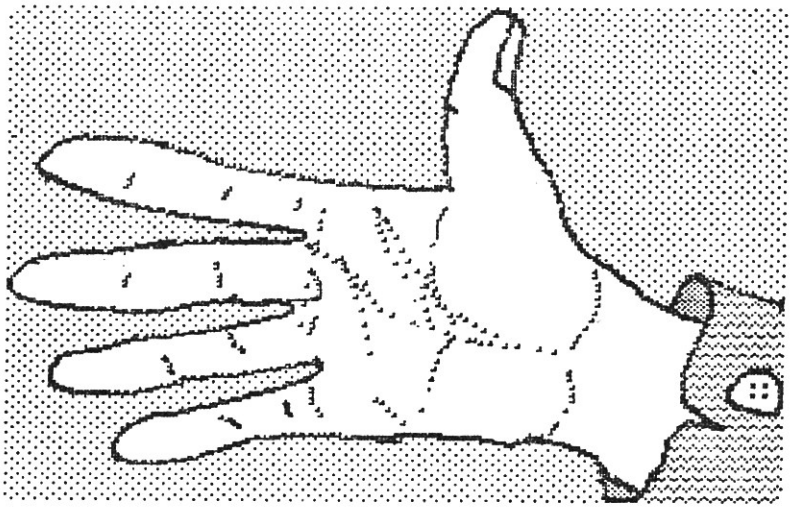
Egy újabb képernyőn figyelemmel kísérhetjük egy program futását. A „szerző" ügyes megoldást alkalmazott, mert mielőtt a program adatot kérne a terminál billentyűzetéről, megjelenik a képernyőn az adatkérő szöveg.

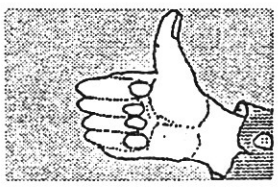
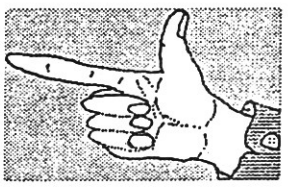
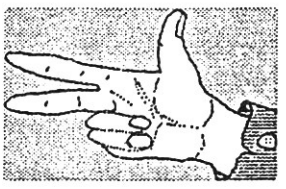
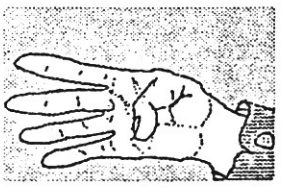
— A párbeszédes környezetben futó programok szerkezetét másképpen kell felépíteni, mint a kötegelt futásra szánt programokét — magyarázza csendben kísérőnk, hogy közben ne zavarja a diákokat. Ha például ezek az adatkérő kiírások nem jelennének meg, a gép az adatra várna, a tanuló meg a gépre, s ha a tanuló nem kap időben észhez, akár órákig is elüldögélhet itt.

Időközben a képernyőn megjelenik egy táblázat: a feldolgozás eredménye. A hallgató az előre kiszámított értékekkel veti össze a képernyőn levőt. Az arcán látszik, hogy a tesztelés eredménye jó.

Haladunk tovább. Az egyik terminálnál valaki éppen befejezi a munkáját. Elköszön a géptől, a gép nyugtázza a kilépést, kiírja az elhasznált központi egység időt, és a kapcsolati időt. A mi időnk is lejárt. Csendben búcsúzunk. Kifelé menet mindnyájan azon töprengünk, hogy a saját személyi számítógépünk környezetében is nagyobb rendet kellene tartanunk.

VAJON MIRE JÓ AZ EGÉSZ ?



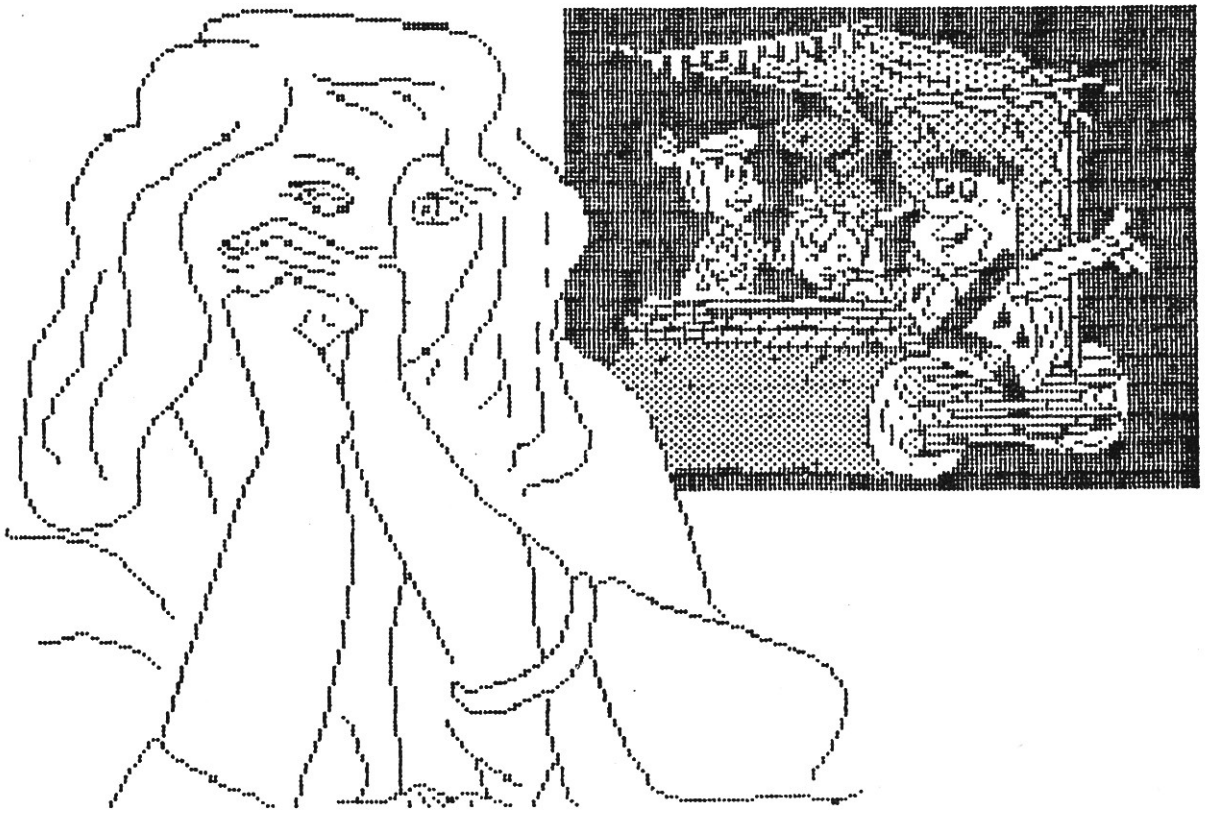





54.

PONT – PONT – VESSZŐCSKE
(Számítógépes grafika)

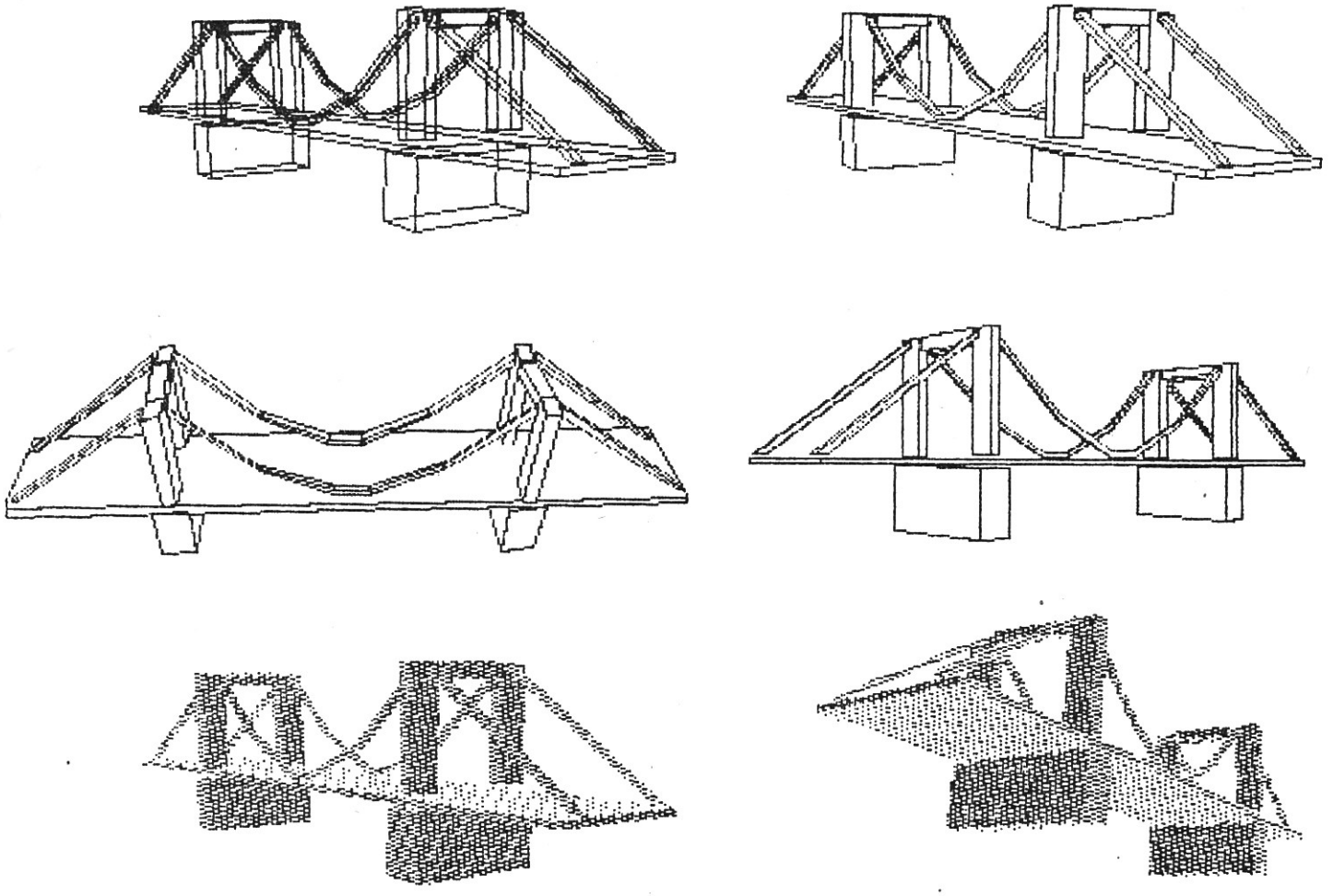
Könyvünk karikaturistája cserét ajánlott fel a szerzőknek: ő megírja ezt a fejezetet, cserében pedig a szerzők illusztrálják. E nagylelkű ajánlattal a szerzők csak részben éltek: a rajzoló írjon, de ők nem rajzolnak . . .

Valószínű, hogy az első számítógépes rajzok azon az egyszerű elven jöttek létre, hogy minden képet pontokra lehet bontani. Igaz, a nagygépek sornyomtatóin ezek a pontok az egyes karakterek (betűk, számjegyek stb.) voltak, vagyis hát pontnak egy kicsit nagyok, de ha megfelelő távolságból néztük, összeállt a figura. Ezek a rajzok úgy készültek, hogy egy képet durva fekete-fehér „pontokra" bontottak, majd ezeket soronként, mint szöveget nyomtatták ki a sornyomtatón. Az ügyesebbek árnyékolni is tudták a képet úgy, hogy az egyes pontokat tömöttebb karakterekkel (pl. * jellel), másokat, a világosabbakat pedig kevésbé tömött jelekkel (x, +, –, /, .) rajzoltak meg. Ezek a képek erősen hasonlítottak a kövekből kirakott mozaikképekre. Ezért szokás ezeket a képeket mozaikképeknek is nevezni, mint ahogyan a mátrixnyomtatót mozaiknyomtatónak is nevezik.

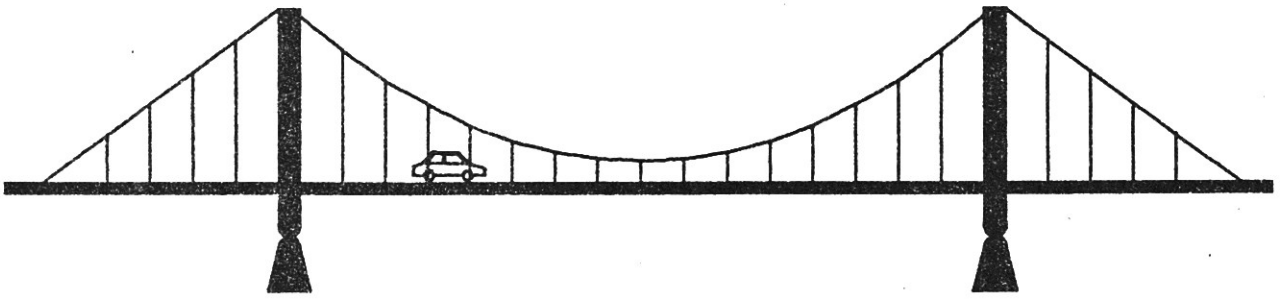


Matisse megfordul a sírjában(?)

A kép pontokra bontása ma is az egyik leggyakoribb módja a számítógépes grafikának. Az elmúlt néhány évben eszközök egész sora – a személyi számítógépek, a nagy felbontású megjelenítők, a rajzoláshoz kifejlesztett legkülönfélébb programok és a számítógépes rajzeszközök – új lehetőségeket teremtettek a képi megjelenítésre.



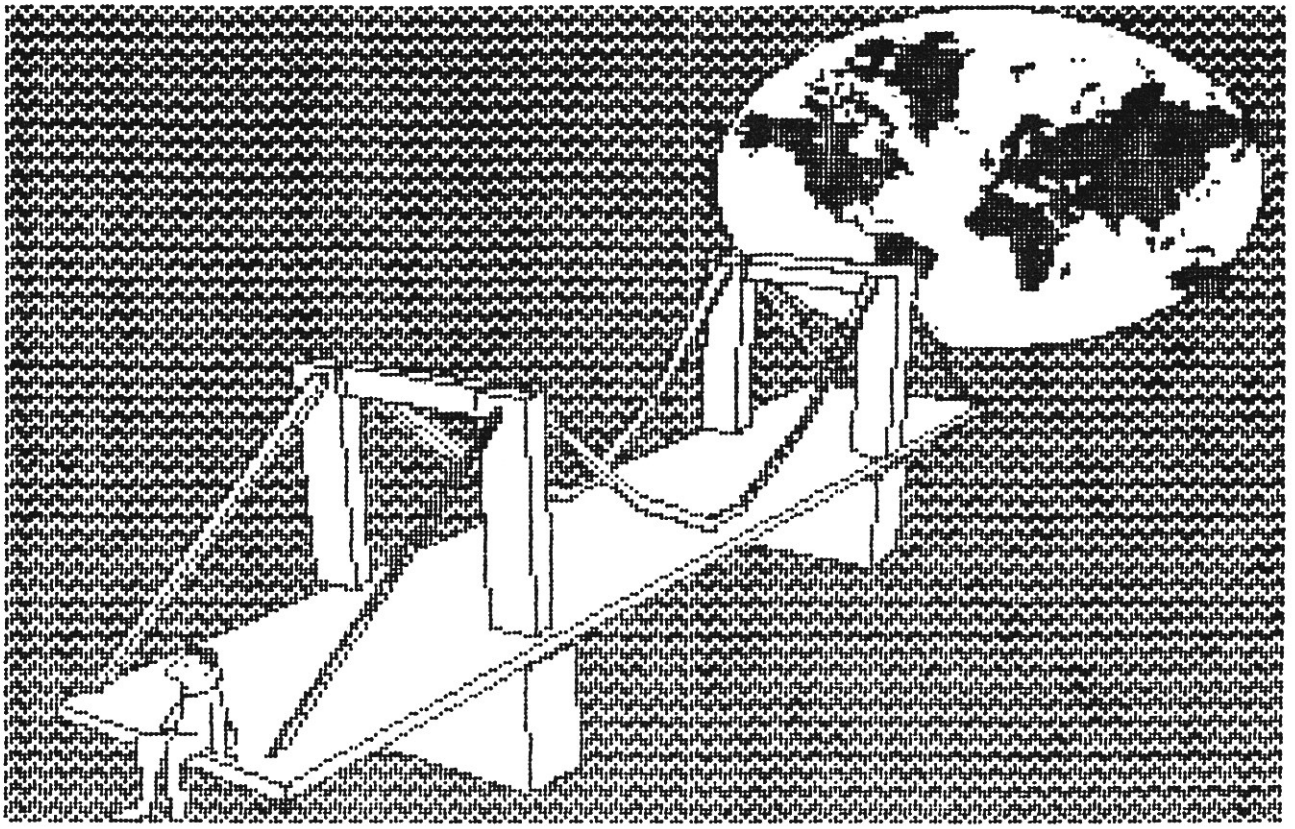
Mielőtt belemerülnénk, meg kell említeni a rajzolás másik módját. A rajzgépek (plotterek) műszaki rajzlapra folyamatos vonalakat húznak. Itt tehát nem mozaikelemekből áll össze a kép, hanem valóságos vonalakból. Ezeknek a programjai nem az egyes pontokat helyezik el a képen, hanem a kiinduló ponttól szögeket, távolságokat, íveket kell meghatározni. Aki nem ismeri a LOGO nyelvet, az úgynevezett teknősbéka grafikát, az is könnyen elképzelheti. Mint karikaturista, ilyen rajzgéppel soha még nem dolgoztam, hiszen ezeket műszaki célokra fejlesztették ki. Példaként azonban álljon itt egy rajzgéppel készült rajz is.



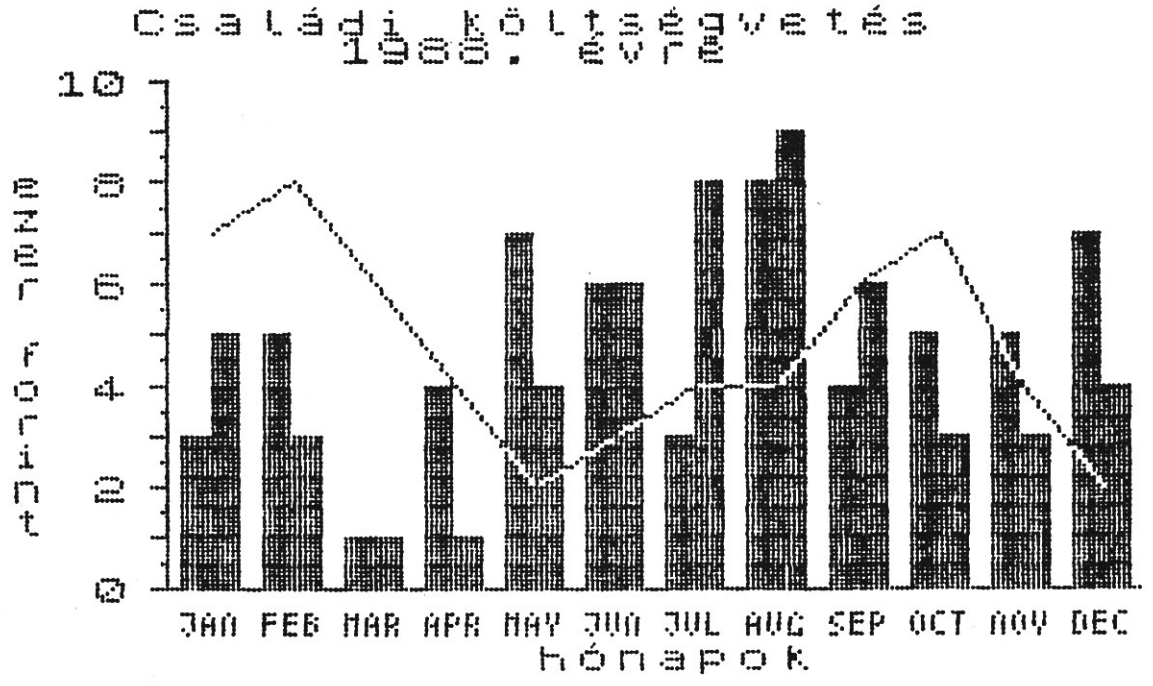
A számítógépes grafikát korántsem csak a művészek játékául, új eszközéül fejlesztették ki. Sokkal praktikusabb oka volt kialakulásának: a műszaki tervezés. A számítógép gyorsan, pontosan tud számolni, segítségével nagyon hatékonnyá tehető a mérnöki munka

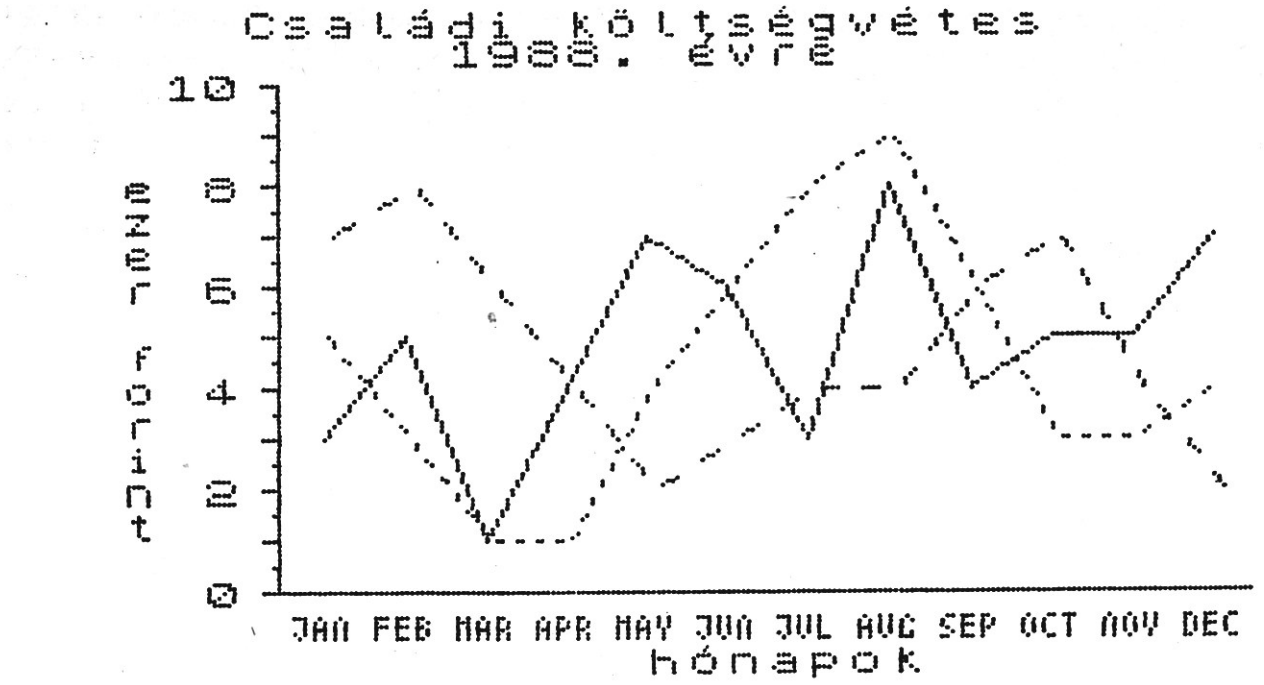
– az eredmények képi megjelenítése nélkül azonban ez a hatékonyság nagyon lecsökkenne, hiszen továbbra is kézzel kellene készíteni a műszaki rajzokat. Az utóbbi években egyre gyakrabban lehetett találkozni ezzel a rövidítéssel: CAD. A három betű jelentése: számítógép támogatta tervezés (Computer Aided Designe). A műszaki grafikai programcsomagok lehetővé teszik a megtervezett dolgok síkbeli (axonometrikus, izometrikus vagy nézeti) megjelenítését, körüljárását, megforgatását is.

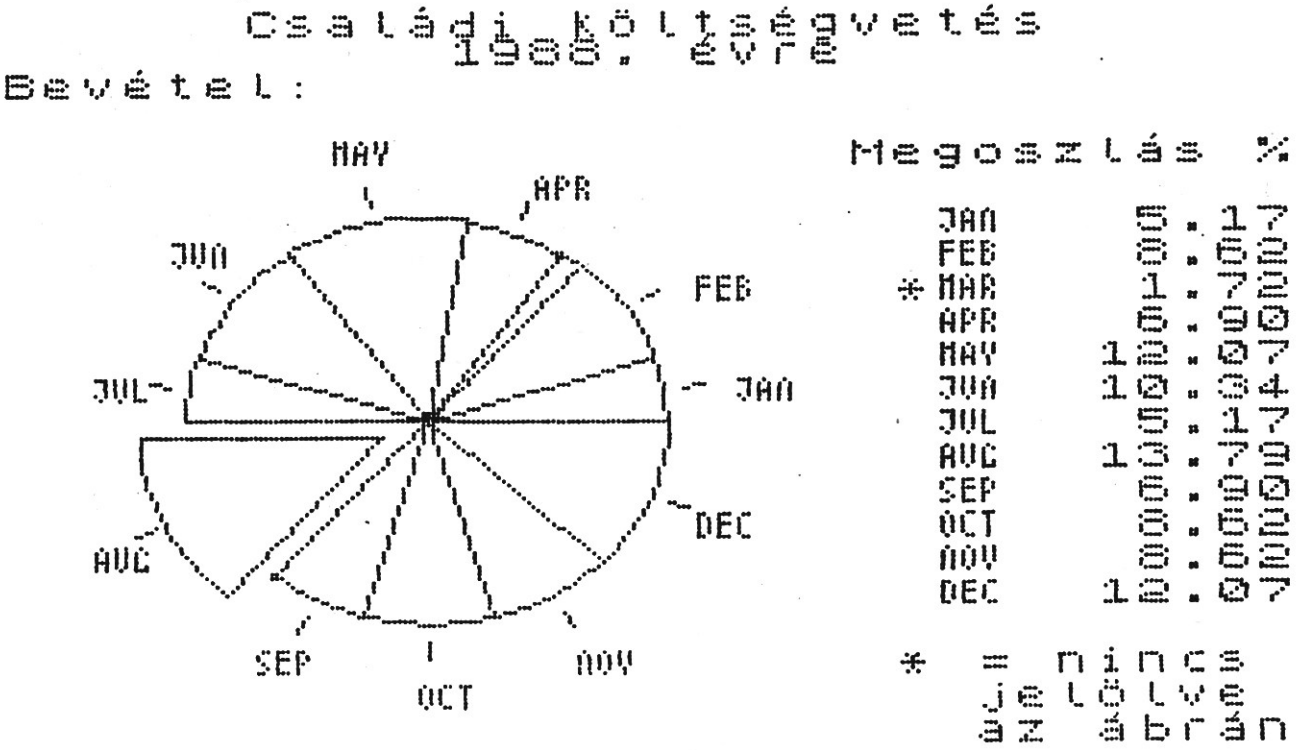
Az itt látható rajzok némelyike mintha nem is műszaki rajz lenne. Nem is az, csak éppen műszaki rajzolói programmal készültek. Aztán, ha a karikaturista elégedett műszaki érzékével, egy rajzába be is építheti eredményét.



Ugyancsak nem a műélvezet igénye hozta létre az üzleti élet grafikáit: a grafikonokat, diagramokat. Lássunk ebből is egy kis ízelítőt:



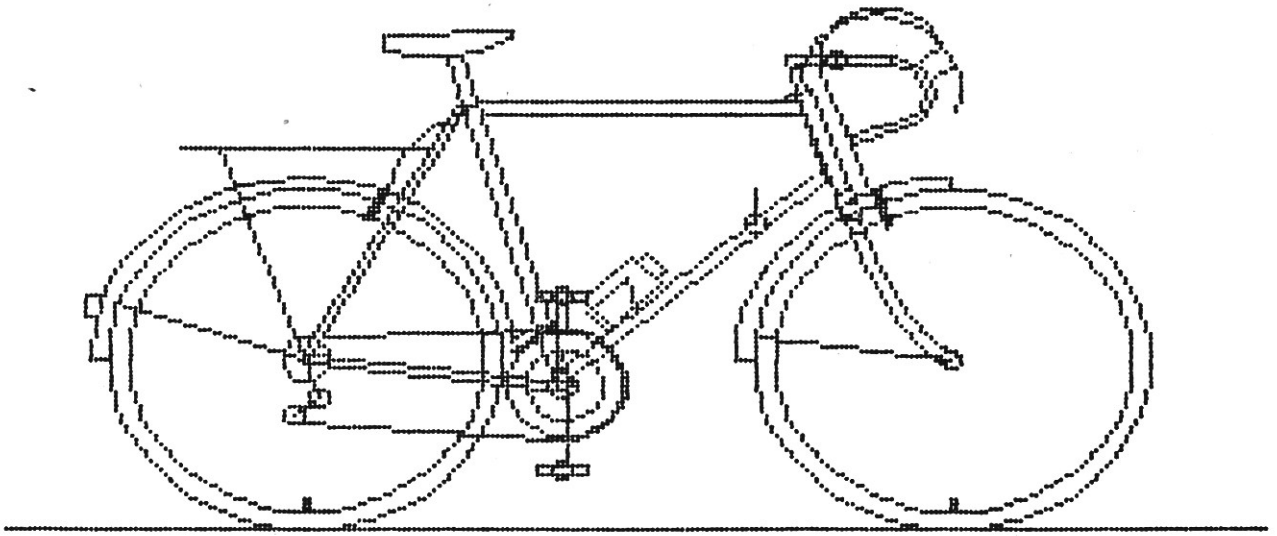




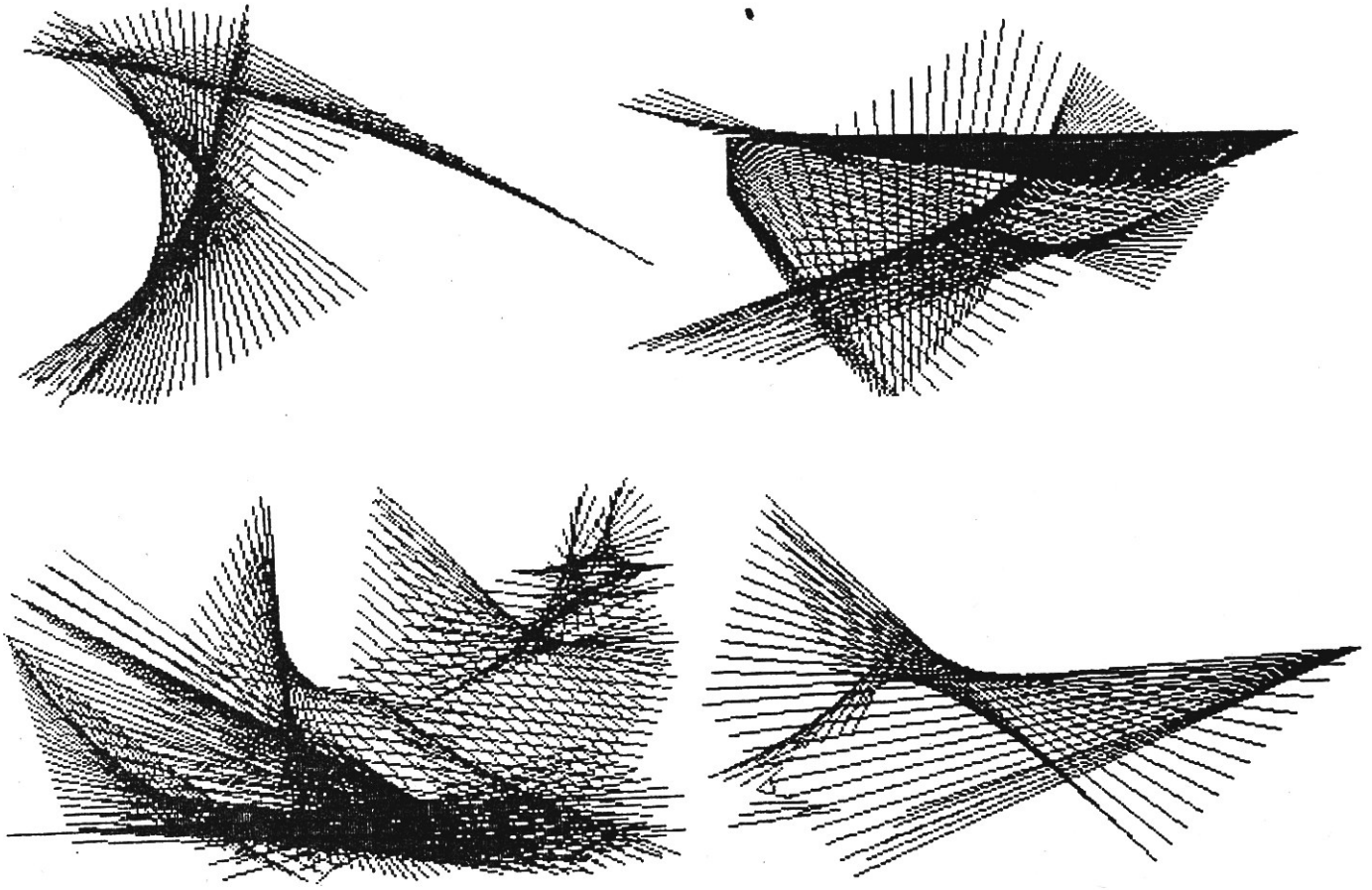
A számítógépes művészi grafika a műszaki rajzolóprogramok, eszközök segítségével alakult ki, ma azonban már külön útjai is vannak. Az „álló" grafika mellett pedig megjelent a számítógépes animáció is. A videoclipek egy része, sok tévéreklám, beúszó főcím számítógéppel készült.

A személyi számítógéppel rendelkezők közt a legismertebb grafika a „szellemgrafika" – a sprite. Ezek az alakok a játékprogramok főhősei, az idegen betolakodók, a szörnyek, a bányarémek, a rablók, a nindzsák, és minden, mindenki, ami és aki csak mozog a játékprogram képernyőjén. E szellemalakok nagysága általában erősen korlátozott, mozgása néhány fázisrajzból áll.

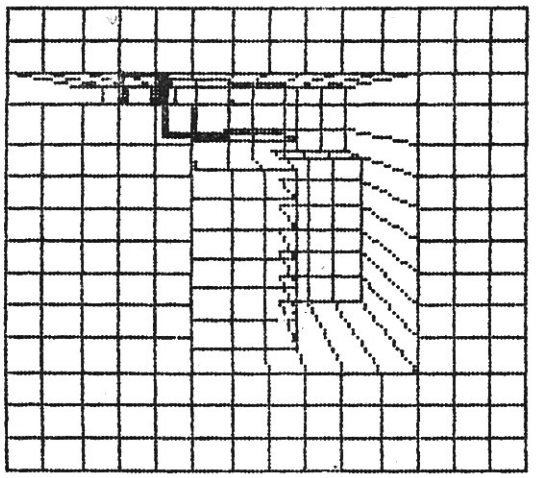
A számítógépes grafika „lelke" a szoftver. A személyi számítógépekhez sokféle, sokat tudó program készült. Legegyszerűbbek a Basic programok. A ZX Spectrum PLOT, DRAW és CIRCLE utasításaival közvetlenül is elkészíthetők bizonyos rajzelemek, illetve kész rajzok is. Ez a kerékpár egy „ismeretlen programozó" műve. Minthogy a programok sokszor másolás útján terjednek, nem tudni, ki volt az eredeti alkotó.

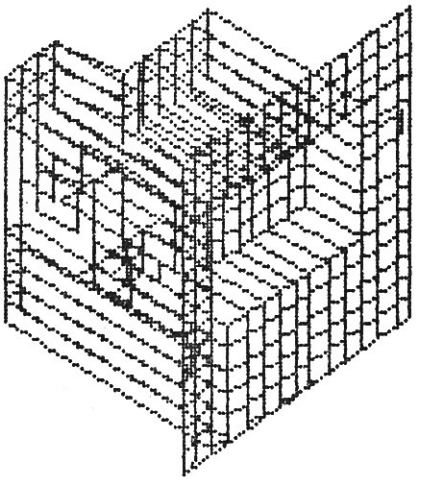


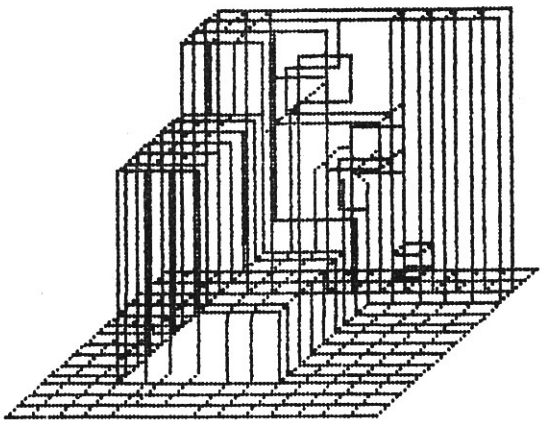
Vannak művészi grafikák, melyek nem egy művész keze, hanem fejében megszületett algoritmus által készülnek. E képek kialakulásában sokszor játszik szerepet a véletlen. Sokan vitatják ezeknek a grafikáknak valódi művészi értékét, hiszen ilyen inkább csak díszítő, a szemnek kellemes látványt nyújtó alkotások belső tartalommal, mondanivalóval nem rendelkezhetnek. Bár a vitában komoly érvek szólnak mindkét oldalon, részemről az utóbbival értek egyet.

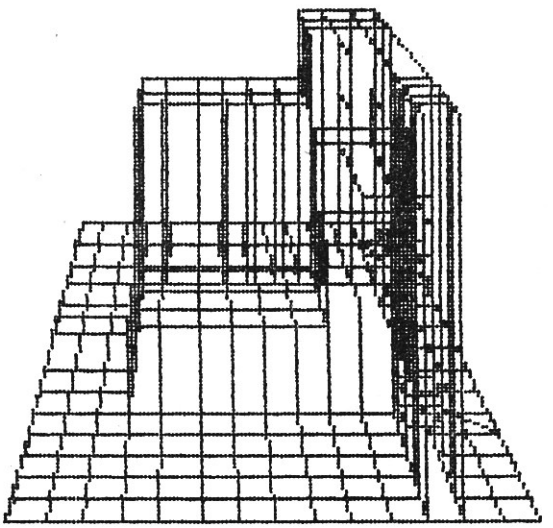


A rajzprogramok ma már a rajzeszközök, lehetőségek egész arzenálját kínálják. A programozók és a grafikusok ötletei szinte kimeríthetetlenek. Nemcsak vonalat húzhatunk, de „vastag ceruzát", „ecsetet", „szórópisztolyt" is használhatunk. Textúrával (mintázattal) tölthetjük ki a megrajzolt felületeket, ablakokat vághatunk a rajzon, az ablakokat sokszorosíthatjuk, kicsinyíthetjük, nagyíthatjuk, torzíthatjuk, átszínezhetjük. Két, vagy több képet is összemásolhatunk, montírozhatunk. 1986-ban magyar rajzprogram lett a nyugati világban az év programja: A CAESAR Stúdió készítette Atari géphez, s ezen például olyan lehetőség is van, hogy egy képen kijelölt kör belső felületét egy gömb felszínére „feszíthetjük", vagy egy téglalap alakú rajzrészletből „hirdetőoszlopot tekerhetünk". Ezek a rajzolóprogramok a grafikus kézügyességét és gépi leleményét igénylik.









Nézetek, vetületek

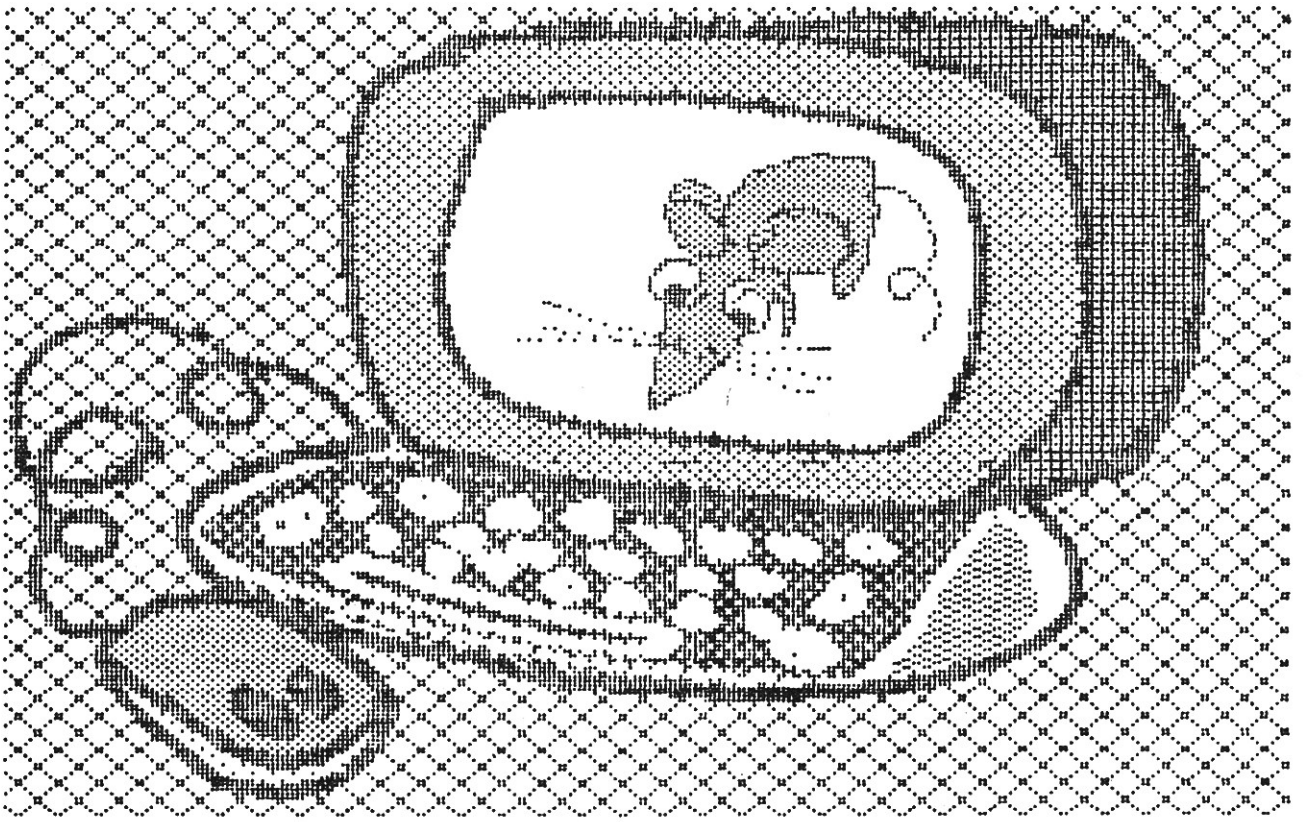
A grafikus szoftver nem egyszerűen a ceruza, vagy az ecset helyettesítését oldja meg. Segítségével olyan művészi alkotások is készíthetők, melyek a hagyományos módon nem, vagy csak nagyon nehezen valósíthatók meg. De ne innen közelítsük meg ezt az új műfajt, vagyis ne azt nézzük, hogyan jeleníthető meg számítógépen a Mona Lisa, és ne is azt, hogy Vasarely képei egyszerű programmal sorra megrajzolhatók lennének: az új eszköz új formanyelvet alakít magának, s nem lesz konkurenciája egy „régimódi grafikusnak", mint ahogyan a fotóművészet sem járatta le a tájkép-, vagy a portréfestőket.

A szoftver önmagában kevés a számítógépen való rajzoláshoz. Az persze, hogy kevés, nem jelenti azt, hogy nem lehet puszta géppel és rajzprogrammal rajzolni, legfeljebb igen nehézkes. Eleinte, mint gyakorló ZX-Spectrumos, magam is a billentyűzet segítségével rajzolgattam. Amikor kipróbáltam a botkormányt, azt hittem, sokkal jobb lesz: csalódtam. Rajzoló ismerősöm viszont esküdött a joystickre (ez a botkormány angol neve).

A számítógépes grafika egyik jó eszköze a fényceruza is. Egy ceruzába fototranzisztort építenek, mely megfelelő illesztő és szoftver segítségével a számítógép megfelelő címére helyezi a képernyőn megjelölt pont x és y koordinátáit.

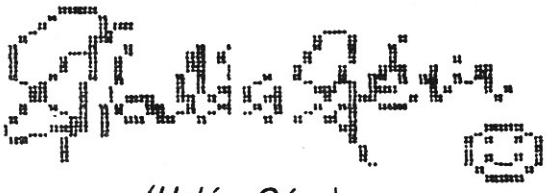
Kedves nevével, jó használhatóságával hamar elterjedt az egér (mouse). Az egér kis (szürke, egér nagyságú) doboz, alján egy gumibevonatú acélgolyóval. Ez a golyó két egymásra merőleges tengelyt forgat, melyek elmozdulásával határozható meg a képernyő egy-egy pontja. Az egeret sima asztallapon tologathatjuk, ezáltal jóval kényelmesebb, mint a fényceruza, melyet a függőleges képernyőn kell mozgatni.

Az itt megemlített rajzeszközök csupán a leginkább elterjedtek, sok másfélével találkozhatunk még. Az elektronikus rajztáblák, rajzasztalok mellett az 1987-es BNV-n már megjelent az az iskolatábla, melyre krétával rajzolhatunk, s a képet akár számítógépes adathordozón tárolhatjuk, akár másológépen közvetlen másolatot készíthetünk róla.



A rajzok megjelenítése a képernyőn kívül nyomtatón is lehetséges. A mátrixnyomtatók többsége úgynevezett grafikus nyomtató, ami azt jelenti, hogy nemcsak karaktereket képes megjeleníteni, hanem tűi egyenként is programozhatók, s így bármilyen kombinációban előretolhatók.

A kötet karikatúrái ZX-Spectrum 48 kbájt kapacitású személyi számítógéppel készültek. A rajzoláshoz egeret használtam és az angliai RAINBIRD szoftverház ART STUDIO programját. A „műszaki" rajzprogramok egyike PSION-termék, neve VU–3D, a másik a német PROFISOFT cég CAD nevű programja.



(Halász Géza)

55.

OLLÓ ÉS RAGASZTÓ NÉLKÜL IS MEGY
(Szövegszerkesztők)

„Klasszikusokat csak pontosan szabad idézni" – idézhetjük pontatlanul e mondat kitalálóját. Aki sokat ír (jelentést, cikket, tanulmányt, esetleg könyvet), az sokszor kényszerülhet arra, hogy másokat idézzen, vagy saját korábbi jegyzeteit, írásait felhasználja. Ennek legegyszerűbb (de egyben legmunkaigényesebb) formája a kiszemelt szövegrészek lemásolása. Ilyenkor jut főszerephez az olló és a ragasztó, amelyek segítségével a felhasználandó anyag (vagy annak fotómásolata) eredeti helyéből fizikailag is kiemelhető és a kéziratba beragasztható.

A technika, pontosabban a számítástechnika az ollót és a ragasztót is nyugdíjba küldte – legalábbis ezen a területen.

szöveg-szerkesztők

Szövegszerkesztőknek nevezzük azokat a programokat, amelyek segítségével a számítógépbe juttatott információkkal szöveg módjára tudunk dolgozni. Ez egyaránt vonatkozik a szöveggel kapcsolatos tartalmi és formai követelményekre. Nem véletlen, hogy egyre több helyen ismerik fel az ilyen rendszerek előnyeit. A szövegszerkesztővel ellátott számítógépek az írógépek helyére települnek.

Vizsgáljuk meg, hogy mi mindenben támogatja a gépíró munkáját egy számítógépes szövegszerkesztő program. Bármilyen szöveg vagy levél begépelése során a képernyőn egyszerűen javítható a készülőfélben levő szöveg: szavak, betűk írhatók át, cserélhetők ki; szavak, sorok törölhetők, újabbak szúrhatók be a megfelelő billentyűk vagy parancsszavak segítségével. Ugyancsak egyszerűen alakítható ki a szöveg formája is.

Korábban elkészített szövegek elővehetők a hajlékonylemezes vagy a winchester tárból, szükség esetén korrigálhatók. Az új változatok természetesen ismét eltárolhatók. Új szövegek könnyen összerakhatók más szövegekből kiemelt részletek, vagy előre gyártott „szövegpanelek" segítségével. Előre elkészített leveleket lehet kiegészíteni és teljessé tenni az éppen aktuális címzett, dátum vagy más adat kényelmes beillesztésével. A címzett a legáltalánosabb levelet is sajátjának érzi néhány személyes utalást olvasva. Az ilyen levelek szövegszerkesztővel történő elkészítése sok gépelési munkától kíméli meg a kezelőt.

A további lehetőségek taglalása helyett inkább próbáljunk meg valami rendszerezést végezni a szövegszerkesztő programok között. Ez nem is olyan egyszerű mint elsőre gondolnánk, mert nagyon sokféle és egymástól is eltérő szolgáltatású programot sorolnak ebbe a kategóriába.

soros editorok

Amióta létezik interaktív programozás (s ez már a hatvanas évek közepén kialakult), vannak olyan programok, amelyek alkalmasak a terminál képernyőjén levő program szövegként való kezelésére. (Ez a feltétele ugyanis a hibajavításnak, ami lehet utasítás törlése, új utasítás beszúrása, betűk, számok javítása.) A legelső szövegszerkesztők (angol nevük szerint text editorok) még csak úgynevezett *soros editorok* voltak, mert egy sort tudtak egyszerre kezelni. Később aztán megjelentek a nagyobb kényelmet adó *teljes képernyős szerkesztők* (angolul full screen editor), amelyeknél a teljes képernyőtartalmat lehetett egyidejűleg kezelni. A személyi számítógépek megjelenésével „felvirradt" a szövegszerkesztő programoknak. Nemcsak azért, mert ugrásszerűen megnőtt a gépek száma, hanem sokkal inkább azért, mert erre a növekedésre alapozva kialakulhatott az a terület, ahol a köznapi értelemben vett szöveg lett a számítógépes feldolgozások tárgya és célja.

teljes képernyős szerkesztők

Jól lemérhető ez a minőségi változás a szövegszerkesztők szolgáltatásain. Lehetővé vált grafikus jelek megjelenítése mind a képernyőn, mind a nyomtatóberendezésen. Új fogalmak születtek, mint pl. a „desktop publishing", amely közel nyomdai igényeket kielégítő színvonalon szerkesztett szöveg előállítását jelenti, vagy a „WYSIWYG", amely az angol "what you see is, what you get" (a képernyőn azt látod, amit kinyomtatva kapsz meg) szavak kezdőbetűi.

A szövegszerkesztéshez ma rendelkezésre állnak:

- célgépek, amelyeknek alapfunkciója a nagyon kényelmesen megvalósítható szöveg-előállítás,
- a szövegszerkesztést biztosító számítógépes programok, amelyek éppúgy futtathatók, mint bármilyen más program.

Annak érzékeltetésére, hogy a számítógépekkel való szövegkezelés mekkora teret hódított már eddig is, egyetlen számadat: az USA szoftverpiacán 1986-ban az összes eladott programoknak már több mint egyharmada szövegszerkesztéssel kapcsolatos.

Milyen szempontok alapján válogathatunk a szövegszerkesztő programok között?

Mindenképpen tudnunk kell, hogy mekkora a tárigénye a programnak. A professzionális kategóriába tartozó gépekhez kapható programok általában 256 kbájt igénnyel lépnek fel.

Nagyobb terjedelmű szövegek (pl. könyv-kézirat) készítésekor fontos tudni, hogy mekkora szövegegységet képes egyszerre kezelni a szövegszerkesztő. Ez a méret általában 64–128 kbájt, de vannak olyan programok, amelyeknél ennél is több.

Bonyolultabb problémáknál az egyidejűleg kezelhető szövegállományok száma, valamint az egyszerre kezelhető ablakok (kijelölt képernyőterületek) száma is fontos lehet. A több ablak egyidejű kezelése nem jelenti azt, hogy ezek az ablakok egyszerre láthatók is, hanem azt, hogy valamely billentyű lenyomásakor az ablak „megnyílik", s esetleg egy előre meghatározott állomány egy része ott megjelenik. Színes monitorok esetében még az is beállítható, hogy milyen színű legyen az ablak háttere.

Fontos szempont lehet a kezelhetőség: pl. mennyire komplikált az egy oldalra írható sorok, az egy sorba írható betűhelyek számának beállítása, az esetleges aláhúzások (s ha a sornyomtató típusa lehetővé teszi), a különböző betűtípusok alkalmazása stb. Az igazán jó szövegszerkesztők ezeket a szolgáltatásokat kényelmes, könnyen megtanulható módon biztosítják.

Vannak olyan szövegszerkesztők, amelyek egy szótár állománya segítségével az elkövetett helyesírási hibákat is észreveszik. Minden országban természetesen az adott nyelv szótára szükséges. S ezzel meg is érkeztünk egy nemzetinek tűnő problémához, az ékezetes betűk használatához. Ezzel azonban már a következő (56.) fejezetben foglalkozunk.

56.

ÉKTELENÜL ÉKES
(A magyar ábécé problémái)

A tv-képújság minden nap közli, hogy mely magyar betűk nincsenek meg a svéd ábécében. De ez csak egy eset a sok közül. Ha végigtekintjük a nemzeti ábécéket, sok olyan betűvel találkozunk, amely az adott nyelv sajátja, és más nyelvekből hiányzik. Ebből a szempontból nem mondható szerencsésnek, hogy a számítástechnika (és így a szövegszerkesztők) alapnyelve az angol, mert ez egy takarékos nyelv, semmilyen ékezetes betű nincs benne. Szöveget viszont mindenki zömmel a saját anyanyelvén akar íratni számítógépen is.

Az ékezet-probléma a hazai számítástechnikában régebbi keletű a szövegszerkesztésnél. Gondoljunk csak az OTP átutalási betétszámlákra, vagy azokra a közüzemi számlákra, amelyeknek elszámolása számítógéppel történik. Ezeknél az elszámolásoknál megjelennek a magyar ékezetes magánhangzók betűképei. Kevesen tudják azonban, hogy mindez csak a nyomtatókon létrehozott változtatás következménye. Vagyis bent a gépben pl. a $ jel belső kódja található (lásd a 15. fejezetet!), mindössze a nyomtató $ betűképét cserélték ki az ö betű képére. (S ugyanígy a többi ékezetes betű is valamilyen speciális jel helyére került.) Ezzel egyidejűleg az adatrögzítő gép klaviatúráján levő $ jelet átragasztották az ö betűre. De az ö betűt tartalmazó nevek például a képernyőn igen furcsán néztek ki. Szegény Töhötöm például Tht$m formában, ami csak akkor érne valamit, ha ezeket a dollárokat zsebre töhötöm, illetve tehetem.

A szövegszerkesztésre használt számítógépeknél az ékezetek használata két problémát vet fel. Az egyik a billentyűzettel kapcsolatos. Minden betűképhez kell egy-egy billentyű. Igaz, hogy a speciális billentyűk felhasználásával (ilyen a CTRL vagy az ALT) egy-egy billentyűhöz több karakter is hozzárendelhető, de ez már a kényelem rovására megy. Nem közömbös a betűk elhelyezkedése sem. Köztudott, hogy az írógépek esetében is létezik a nemzeti karakterek elhelyezésén túl egy amerikai, egy német és egy francia elrendezés (az első a QWERTY elrendezés, ilyen sorrendben következnek a második sor betűi, a német rendszerben az y és a z helye fel van cserélve — ezen alapszik a magyar írógép-billentyűzet is, a franciánál pedig AZERTY — itt az a a q-val, a z a w-vel van felcserélve). A számítógépek billentyűzete általában az IBM szabványt követi, de pl. Franciaországban követi a francia írógép billentyűzetet még abban is, hogy itt a számokat „nagy betű" állásban lehet leütni. Ha ékezetes betűket is akarunk, ezeknek helyet kell biztosítani. (Ilyen pl. a Proper 16 géphez forgalmazott UNIBOARD billentyűzet. Profi számítástechnikusok nem szeretik, mert nagyon eltér az általuk megszokott IBM szabványtól.)

A másik probléma a betűképet megjelenítő ROM-mal kapcsolatos. A hiányzó betűk ritkán használatos jelek helyére égethetők be, a ROM cseréje árán, mivel annak is korlátozott a kapacitása.

Léteznek olyan programok, amelyek segítségével a billentyűzet átdefiniálható. Ez persze csak a ROM-ban létező karaktereknek a billentyűzeten elfoglalt helyét változtatja, de sokszor ez is sokat segíthet. Ilyenkor persze a billentyűzetet is át kell ragasztgatni, vagy új „sapkákat" kell húzni a megváltozott funkciójú billentyűkre.

Általános elvként kimondható, hogyha a felhasználó hajlandó bizonyos kompromisszumokra, és például nem ragaszkodik az írógép klaviatúra esetleg megszokott elrendezéséhez, felvállal az ékezetes betűknél némi kényelmetlenséget, a magyar ékezetek problémája mindig megoldható.

Ugyanakkor egyes nagy szoftvergyártók célul tűzték az európai nemzeti ábécéket tartalmazó karaktermegjelenítő ROM tárak kidolgozását. Találkozhatunk itthon is olyan törekvésekkel, amelyek célja, hogy a már meglevő szövegszerkesztőket alkalmassá tegyék a legkülönbözőbb karakterek kezelésére. Ilyen hazai eredmény a Commodore 64 gépre készített Easy Script szövegszerkesztő magyar betűelrendezésű változata.

Végezetül az ékezetek kapcsán hadd mutassunk meg egy ékes problémát, amelynek számítógéppel való megoldása sokkal bonyolultabb, mint a manuális. Az ábécé szerinti rendezésről van szó, olyan környezetben, ahol magyar ékezetek is előfordulnak. (Tehát például magyar neveket kell rendezni.) A probléma kettős. Egyrészt a korábban elmondottak szerint az ékezetes magánhangzók a használaton kívüli speciális jelek belső kódját „bitorolják". A speciális jelekhez tartozó belső bitkombináció helye a kódtáblázatban a betűk után van, vagyis a rendező program nem az általunk kívánt helyre tenné az ékezetes magánhangzókat. Ennek megakadályozása több módon lehetséges, ezzel most nem foglalkozunk, csak a problémára hívjuk fel a figyelmet. A feladat azonban ezzel még nincs megoldva. Tudvalevő, hogy a magyar ábécébe történő rendezés szabálya szerint a hosszú és rövid magánhangzók azonosnak számítanak. Tehát pl. a „péksütemény" az ábécében megelőzi a „pemetefü"-vet, mert az é és az e egy, tehát a k betű miatt fog a péksütemény a pemetefű elé kerülni, hiszen annak a harmadik betűje m.

A számítógépes rendező programnak tehát egyrészt észre kell vennie, hogy az é és az e hang (betű) egymástól különböző, másrészt tudnia kell, hogy mégis azonosak. Éktelen zűrzavarhoz vezethet mindez, ha e feladat programozója nem ékessége szakmájának.

57.

MINDENT EGY HELYEN
(Adatbázisok, adatbankok)

A mai kor emberére szakadatlanul újabb és újabb információk tömege zúdul. A nap 24 órája sem volna elegendő arra, hogy egy szakember elolvashassa a szakterületén fellelhető új publikációkat. Hetente vagy havonta rendszeresen megjelenő vastag szakfolyóiratok százait kellene állandóan szemmel tartania, ha mindenről részletesen értesülni akarna, ami a saját kiválasztott érdeklődési körén belül történik. Nem véletlenül mondják azt, hogy manapság egy találmányról sokkal nehezebb megállapítani, hogy nem valósították-e meg már máshol, mint ténylegesen kiötleni. De nemcsak a részterületek problémáit megoldó szakembereknek vannak gondjaik az információk rendszerezésével. Az állami és a gazdasági élet vezetőinek is annyiféle jelentős tényezőre kell tekintettel lenni döntésük meghozatalánál, hogy ezek összegyűjtése, áttekintése, hatásának előre való felmérése ma már meghaladja a legjobb emberi agy képességét is. E nagy tömegű adat tárolására, rendszerezésére, az adatok közti kapcsolatok megállapítására vonták be napjainkban a számítógépet. A számítógépek nagy „emlékezőképessége" és gyorsasága lehetővé tette az ember agyának, gondolkodóképességének kiterjesztését, bizonyos mértékű helyettesítését. Ez természetesen nem jelenti azt, hogy a számítógépek fognak rendszeresen gondolkodni és dönteni az emberek helyett. A gépek általában csupán az adatokat fogják ehhez szolgáltatni.

Tizenöt-húsz évvel ezelőtt a számítógépes feldolgozáshoz felhasznált adatállományok szervezése mindig egy konkrét feldolgozáshoz igazodott. Ez például egy raktárkészlet nyilvántartásánál azt jelentette, hogy minden anyagnak volt egy azonosító cikkszáma, melynek alapján — az adattár szervezésétől függően (lásd a 44. fejezetet!) — közvetlenül, vagy az előtte levők elolvasása után sorosan meg lehetett találni az erre a cikkre vonatkozó információkat (a cikk ára, mennyi van belőle a raktáron, mennyi a minimális készlet, hol rendelhető meg, milyen szerkezeti elemekhez hány darab kell belőle stb.). Az adatokat még csoportokba is lehetett rendezni, és ezekre vonatkozóan összesített információkat is lehetett kapni. Így például egy gépkocsigyárban az egyes szerkezeti egységek között lehetséges a következő durva felosztás: kerekek, motorok, karosszériák, erőátviteli művek, elektromos berendezések stb. és ezeken belül minden tételt tovább bonthatunk. Például a kerekek gumiabroncsból, szelepből, keréktartó házból, dísztárcsákból és rögzítő csavarokból tevődnek össze. Ezen belül az abroncsok feloszthatók tömlős és tömlő nélküliekre, ezek aztán radiál és normál köpenyre, majd fajtán belül méret szerint is osztályozhatók, míg végül eljutunk a gyártás szempontjából elemi egységnek tekinthető alkatrészekig. (Az, hogy melyek az elemi egységek, függhet a gyártás folyamatától. Ha pl. a motorok teljes összeszerelését is az adott gyár végzi, akkor a motor egyes darabjait, csavarjait tekintjük elemi alkatrésznek. Ha viszont az egész motort egyben vásárolja meg a gyár valamelyik kooperáló vállalattól, és azt egy az egyben beépíti, akkor a teljes motort tekinthetjük az elemi alkatrésznek, az összetevők további részletezésére nincs szükségünk.) Problémát okoz azonban ennél az adatszervezési módnál, ha azt kívánjuk megtudni, hogy van-e elég alkatrész adott számú, meghatározott típusú gépkocsi legyártásához. Ehhez

előbb tudnunk kell, hogy milyen elemekből épül fel az adott típusú gépkocsi, majd az egész más rendszer szerint tárolt alkatrész-adatokból kell keresni az ezekre vonatkozó készletadatokat. Ez igen időigényes, mert gyakorlatilag minden egyes elemi adatot át kell nézni, hogy nem kell-e a megadott típusú gépkocsiba. Éppen ezért régebben ezt úgy oldották meg, hogy ugyanazon elemi adatokat ennek a feldolgozásnak megfelelő szervezettséggel még egyszer tárolták. Ekkor az adatok nem a gyártóegységeknek megfelelő hierarchiában, hanem a „mi mihez szükséges" csoportosításban szerepeltek. Természetesen bármely anyag felhasználásakor mindkét adatállományt módosítani kellett.

adatbázis, adatbázis-kezelő rendszer

A mai adatbankok megszüntették a többszörös adatállományokat. Közös *adatbázist* építettek ki, melyekben minden adat csak egyszer szerepel, de ezek olyan módon vannak egymással összeláncolva, hogy bármilyen ésszerű feldolgozás céljára megfelelő módon elérhetők és csoportosíthatók legyenek. Ennek érdekében az adatbázis nemcsak az adatokat, hanem a köztük fennálló logikai kapcsolatokat is tartalmazza. A fizikailag különböző helyen tárolt, de logikailag összetartozó adatok összegyűjtéséről egy külön rendszer, az *adatbáziskezelő rendszer* gondoskodik. Az adatok zöme (esetenként esetleg több százmillió vagy milliárd adat) közvetlen elérésű mágneslemezen van, míg az adatbáziskezelő program működő része és az éppen feldolgozásra kerülő adatok a központi tárolóban vannak. Természetesen a közös adatbázishoz több felhasználó is hozzáférhet, sokszor egyidejűleg is. Ezért a legtöbb adatbáziskezelő rendszer lehetővé teszi a párbeszédes üzemmódot is. A felhasználók terminálon keresztül teszik fel a számítógépnek a kérdést vagy adják meg a teendőket, a válasz vagy a feladat elvégzésének nyugtázása rövid időn belül megjelenik a képernyőkön. Ezek a kérdések vonatkozhatnak egy elemre vagy egy elemcsoportra, de érinthetik az adatok egészét is, vagyis teljes körű feldolgozást is kívánhatnak a géptől. Ez utóbbi esetben rendszerint valamivel hosszabb idő után, kinyomtatott formában kapjuk meg az eredményt.

A kérdések és kérések lehetnek egészen egyszerűek, például olyanok, hogy van-e hely egy adott repülőjáratra, és ha igen, akkor azt foglalják le X. Y. részére. Lehetséges azonban a kiválasztás összetettebb feltételek alapján is, pl. egy személyi adatbanktól megkérdezhetjük, kik azok a 30 és 40 év közti nőtlen férfiak, akiknek vegyészmérnöki diplomájuk van és angolul és oroszul is legalább középfokú szinten beszélnek. A kérdések szabatos megfogalmazására külön nyelv szolgál, mely vagy egy önálló, az adott adatbáziskezelő rendszerhez tartozó nyelv, vagy pedig valamely közismert programozási nyelv, melybe az adatbáziskezelő rendszer beépíthető.

A működő adatbankok zöme az ipari termelésirányítás, a bankszakma, a személyi nyilvántartás és a népgazdasági információs rendszerek igényeit elégíti ki, mivel itt az adatok mennyisége, az információs igények gyakorisága és a pontos információk fontossága indokolttá teszik a meglehetősen drága műszaki berendezések használatát és az adatbank felépítésébe és üzemeltetésébe fektetett nagymértékű szellemi munkát.

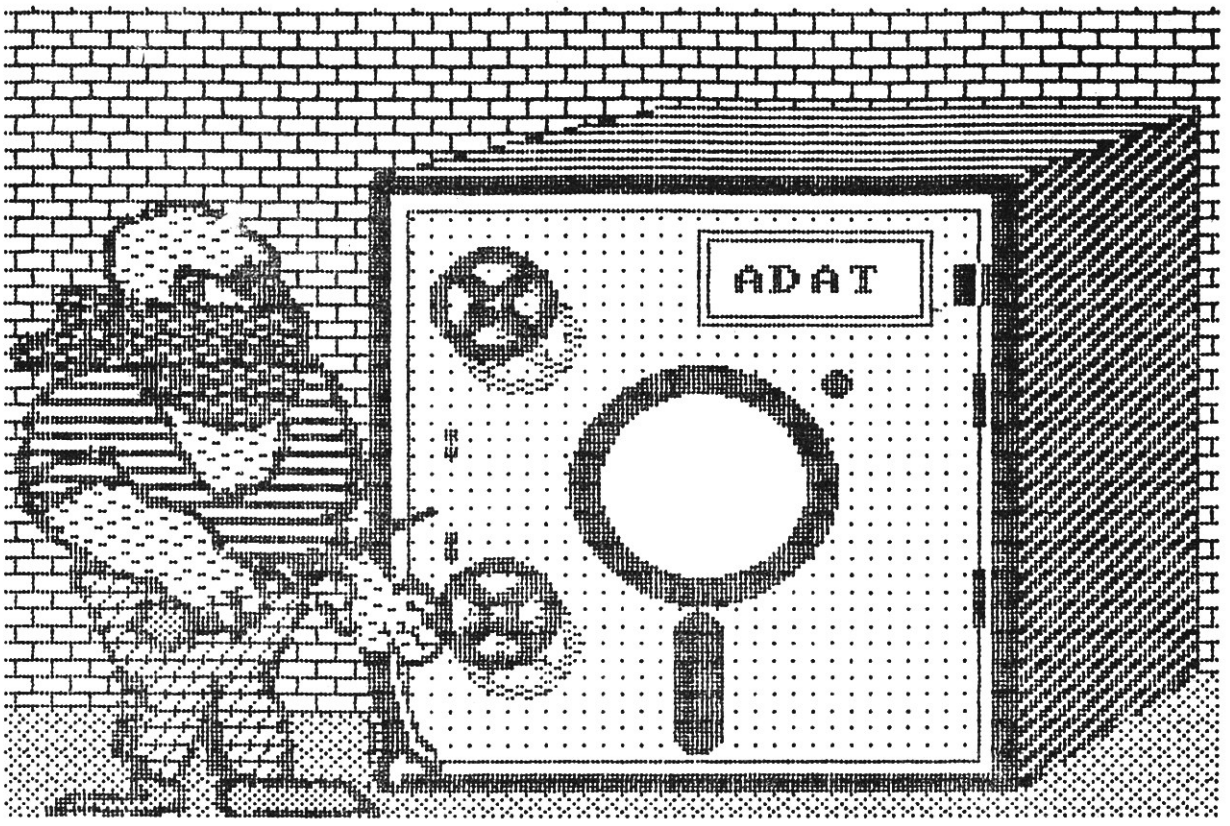
Mivel a közös adatbázist egy helyen tárolják (pl. pénzügyi adatbanknál rendszerint a bank központjában), de az információra több helyen sürgősen szükség lehet (pl. a kerületi fiókokban), ezért a lekérdező helyek egy távadatfeldolgozó rendszer részei, melyet telefonkábelek kötnek össze a központi géppel. Ezáltal az adatbank szolgáltatásai nincsenek helyhez kötve, a viszonylag nem drága adatvégállomás üzembe állításával bárhonnan igénybe vehetők.

A közös adatbázishoz az adatvégállomásokon keresztül bárki hozzáférhet. Ezért különös gondot kell fordítani az adatok titkosságának és sérthetetlenségének biztosítására. (Lásd a 39. fejezetet!) Ez azt jelenti, hogy nem mindenki kaphat információt a teljes adat-

bázisról. Az adatok különböző szintű csoportjait jelszavak védik és az így védett információkhoz csak azok férhetnek hozzá, akik ezt a jelszót közlik a számítógéppel. Külön lehet szabályozni azt is, hogy bizonyos jelszóval csak olvasni lehet az adatbankból, vagy egyes részéből, míg más jelszó ismeretében meg is lehet változtatni az ott tárolt adatokat. Mindezeket az ellenőrző műveleteket az adatbázis felépítése után az adatbáziskezelő rendszer automatikusan elvégzi.

Magyarországon a két leggyakrabban használt adatbáziskezelő rendszer az IDMS és a dBASE. Az *IDMS*-t nagyszámítógépekre, nagy, bonyolult szerkezetű adatállományok kezelésére dolgozták ki. Alkalmazása komoly számítástechnikai tudást, profi programozási és rendszerszervezési ismereteket követel. A *dBASE* személyi számítógépeken futtatható. Felépítése annyira illeszkedik a felhasználók logikájához, hogy minimális számítástechnikai, programozási alapismeretekkel is igen gyorsan megtanulható, és ki-ki igen gyorsan felépítheti és lekérdezheti vele saját igényei szerint kialakított adatállományát.

IDMS, dBASE



A számítógépes információs rendszerek elterjedését az információs igények növekedése, a műszaki fejlődés és gazdaságossági okok is elősegítik. Bár az nem igaz, hogy a gép soha nem téved, hiszen műszaki hiba bármikor előfordulhat, de a számítógép belső ellenőrzési rendszere annyira megbízható, hogy ez a hibaarány nagyságrendekkel jobb bármilyen más adatszolgáltatási rendszer hibaszázalékánál. Döntő érv lehet a számítógép mellett az időtényező is, hiszen kézi nyilvántartásnál esetleg napokba telhet a szükséges adatok előkeresése, míg a gép ugyanezt perceken belül el tudja végezni. Ez természetesen akkor szempont, ha az információra sürgősen van szükség.

Ezenkívül a számítógépes adatbank még viszonylag olcsó is. Ha egy tisztviselő, akinek feladata csupán az adatok szolgáltatása, percenként átlagosan 5 adatot tud előkeresni és a fizetése havi 6000 Ft, akkor egy információ 10 fillérbe kerül. Egy közepes nagyságú adatbázissal dolgozó számítógép gépórája 15 ezer forint körül van. Egy ilyen gép akár ezer adatot is tud szolgáltatni másodpercenként, így egy információ költsége 0,4 fillér lenne.

A gép tehát huszonötször olcsóbb a hivatalnoknál. Még tovább javul a helyzet a gép javára, ha nemcsak előkeresni kell az adatokat, hanem számolni is kell velük, mielőtt a végleges információt előállítjuk belőlük. (Pl. több adat átlagát, különbségét vagy egymáshoz viszonyított arányát kell megadni.) Ráadásul a gépnek nem jár fizetett szabadság, ebédidő, és ha néha mégis gyengélkedik (műszaki hibája van), abból is rövidebb idő alatt épül fel, mint a tisztviselő az influenzából.

Mielőtt azonban abba a tévhitbe ringatnánk magunkat, hogy a jövőben mindent a számítógépes adatbankokból fogunk megtudni, néhány megjegyzést kívánunk tenni, hogy lássuk e rendszerek korlátait is. Az első, amit figyelembe kell vennünk az, hogy a korábban felsorolt gazdasági előnyök csak látszat-előnyök akkor, ha nincs szükségünk ilyen nagy mennyiségű információra. A második megjegyzésünk az, hogy a túl sok információ gyakran csaknem ugyanolyan veszélyes lehet, mint az információ hiánya. Az a vezető, akit az elemző, a számítógép által összefoglalt adatok helyett az ugyancsak a gép által szolgáltatott részletes adatok tömegével terhelnek meg, könnyen elveszhet a részletekben. A fontos kérdések helyett olyanokkal kénytelen foglalkozni, olyanokban dönt, melyeket beosztottjai is el tudnának intézni, míg a fontos problémák idő hiányában elintézetlenek maradnak.

Végül még egy veszélyre szeretnénk felhívni a figyelmet. Ne higyjünk száz százalékosan a számítógépek csalhatatlanságában. A mai gépek ugyan szinte sohasem tévednek (és ha mégis, akkor annak, hogy ezt nem vesszük észre, kizárólag emberi okai vannak), ezért bármilyen hiba, ami a gépi információszolgáltatásból adódik, sokkal komolyabb következményekkel járhat, mint hogyha azt emberek adták volna, ahol a tévedés lehetőségével egy kicsit mindig számolni kell. Ennek ellenére a mai kor emberének szüksége van a számítógépes adatbankokra, melyeket néha közvetlenül, máskor csak közvetve, újságokon, televízión keresztül kapott információk formájában használunk.

58.

SZÉP IS LEGYEN, JÓ IS LEGYEN!
(Dokumentum-visszakereső rendszerek)

A család együtt ünnepel. Természetesen előkerülnek a régi fényképalbumok is. — Mutasd meg a Pistának, milyen édes volt a Maris négyéves korában, amikor Dunabogdányban nyaraltunk és a Julissal együtt lefényképeztük őket, ahogy kavicsokat dobáltak a vízbe — mondja a nagymama a nagypapának.

De hiába forgatják végig az összes fotóalbumot, sehol nem találják ezt a képet. Találnak viszont helyette három olyat, ahol nyelvet öltenek egymásra és szamárfület mutogatnak a lányok, tehát egyáltalában nem édesek. — Pedig biztos meg kell lennie! Pontosan emlékszem rá! Ha öreg is vagyok már, az én fejem olyan, mint egy modern kompjuter — dohog a nagymama.

Sajnos mégsem olyan. Egy számítógép (és nem kompjuter, ahogy a nagymama mondja), amelyiket ezzel a feladattal bíztak volna meg, és természetesen az ehhez szükséges programokat és adatokat is bevitték volna a memóriájába, mágneslemezeire, azonnal közölte volna, hogy ez a kép az aranyfedelű családi album hetedik oldalán a legbelső sorban található balról a második helyen és nemcsak a Julis és a Maris van rajta, hanem a Bodri kutya is, amint ugat, — bár az már nem hallható.

dokumentum formátumú; tényadat formátumú adatbázisok

Az adatbázisokat, az információs szolgáltatáshoz felhasználandó adatoknak és a köztük levő kapcsolatoknak az összességét az előző fejezetben tárgyaltuk. A családban az előbbi példában hiányolt számítógépnek szintén információszolgáltatás volna a feladata, de funkcióját tekintve kissé más módon, mint az adatbankoknál. Az információ tárolásának és felhasználásának módja szerint különbséget teszünk *dokumentum formátumú* és *tényadat formátumú adatbázisok* között. Az előbbi részben, vagy teljes egészében önmagában is értelmezhető folyamatos szövegekből áll, melyből a megfelelő dokumentumokat a szövegben levő szavak, szócsoportok azonosítása alapján választjuk ki. Az utóbbiban az adatok egyértelműen meghatározott helyeken, adatmezőkben (lásd a 43. fejezetet) helyezkednek el, és a kiválasztás ezen mezők és tartalmuk azonosítása révén történik. A szokásos szóhasználatban csak utóbbiakat szokás adatbázisnak nevezni. A dokumentum formátumú adatbázisokat kezelő rendszereket megkülönböztetésül *szövegesinformáció-kereső rendszereknek* hívják. (Angol elnevezésük, Document Retrieval Systems után elfogadott még a „dokumentum visszakereső rendszerek" elnevezés is.)

szöveges-információ--kereső rendszerek

Az adatbázisban egy adott kérdésre válaszként kapott eredmények általában új adatok (rekordok), melyeket az adatbázisban meglevő különböző rekordokból állít össze a kívánt szempontok alapján az adatbáziskezelő rendszer. A szöveges információ visszakereső rendszerekben a kérdésre válaszként kapott rekordok megegyeznek az adatállomány azon rekordjaival (illetve annak egy meghatározott részével), melynek mezői részben vagy teljesen tartalmazzák a kereséskor megadott jelsorozatokat és ezek között a kérdés feltevésekor megszabott logikai kapcsolatok is teljesülnek. Tipikus példa erre, amikor cikkekből, könyvekből a tartalmuk rövid leírása alapján, külön tárgyszavazás nélkül akarjuk kiválasztani a bennünket érdeklő anyagok leírását, mondjuk azokat, amelyeknek ismertetésében

szerepel a „népszerű" vagy „ismeretterjesztő", és ezen kívül a „számítástechnika" szó is (pontos meghatározása a jelsorozatoknak), vagy olyan szavak, melyek a „népszerű . . ." vagy „ismeretterjesztő . . ." valamelyikével, és a „számítástechnika . . ." jelsorozattal kezdődnek. Az előbbi esetben azok a művek, melyeknek ismertetésében ez szerepel: „. . . népszerűen ismerteti a számítástechnikának a legújabb eredményeit . . ." nem fognak megjelenni az eredményrekordok között, az utóbbiban viszont igen.

A szöveges információ visszakereső rendszerekben a keresés és egyeztetés történhet az egyes dokumentumok soros végigolvasásával. (A legtöbb szövegszerkesztő — lásd az 55. fejezetet! —, így működik.) Ez nagyobb adatállománynál már igen hosszú időt vesz igénybe. Ezért a szövegesinformáció-kereső rendszereknél más módszert alkalmaznak. A tárolandó szöveg minden egyes szavára, vagy az anyagoknak a rendszerbe való bevitelekor előre megjelölt szavaira egy invertált file-t építenek fel (lásd a 44. fejezetet!), és a keresés ennek segítségével valósul meg. Ekkor természetesen az indextáblák felépítése miatt a tárolóigény lényegesen nagyobb (a gyakorlatban — a keresésre megjelölt szavak számától függően — általában mintegy 2—3-szoros) lesz. A dokumentum bevitele, módosítása lényegesen hosszabb időt vesz igénybe, de a visszakeresés sokkal gyorsabb lesz, mint a soros végigolvasással kereső rendszereknél.

Az indextáblázat kulcsrésze a dokumentumoknak a keresésre szolgáló szavait tartalmazza, míg az index azoknak a dokumentumoknak az adatbázisbeli azonosítóját (címét), és ezen belül annak a mezőnek a sorszámát, melyekben a kulcs előfordul. Egyes rendszerekben szerepel még a kulcs mezőbeli relatív helye is, mely lehetővé teszi elsősorban összetett keresési feltételeknél azon találatok megkülönböztetését és kiválasztását, melyeknél a keresett jelsorozatok egymáshoz viszonyított helyzete nem közömbös. (Pl. a Péter László nevű személyre való utalást tartalmazó dokumentum keresésekor a keresőkulcsok „Péter" és „László", de előfordulásuk sorrendje lényeges, a „László Péter"-re hivatkozó anyagokra nincsen szükségünk.) A dokumentumok kiválasztása a kereső szavak alapján ezeknek az indextáblázatban való kikereséséből és a hozzájuk tartozó címeken levő dokumentumok beolvasásából áll. Természetesen ahhoz, hogy egy ilyen rendszer a felhasználó számára használhatóan működjék, szükséges, hogy a keresésre szolgáló szöveg (elsősorban a tartalmi leírás) ne csak szakmai, hanem számítógépes keresési szempontból is megfelelő legyen. Ezt segíti elő a csonkított alak alapján való keresés. Ilyenkor a keresőszóval való egyeztetés csak egy speciális karakterig (például a kereső szóban megadott $ jelig) történik. Ilyen módon ugyanannak a szónak a különböző ragozott alakjait nem kell külön keresnünk, a SZÁMÍTÁSTECHNIK$ szó keresésével megtaláljuk az összes olyan dokumentumot, melyben a SZÁMÍTÁSTECHNIKA, SZÁMÍTÁSTECHNIKAI, SZÁMÍTÁSTECHNIKÁVAL, SZÁMÍTÁSTECHNIKUS, SZÁMÍTÁSTECHNIKUSOK . . . stb. szavak szerepelnek.

szinonima-szótár

További segítséget nyújt a felhasználónak a *szinonima-szótár.* Ez egy önálló állomány, melynek keresési kulcsa az alapszó (pl. BENELUX), mezői a vele egyenértékű vagy azt helyettesítő szavak (pl. BELGIUM, HOLLANDIA, NÉMETALFÖLD, LUXEMBURG). Amennyiben a keresést a megadott szó összes helyettesítőjére, szinonimájára is ki akarjuk terjeszteni, akkor ezt a rendszerrel egy speciális kifejezés (például ANY = bármelyik) megadásával közöljük. Ebben az esetben a rendszer automatikusan végignézi a szó mindegyik szinonimájához, mint kulcshoz tartozó indexet az összes dokumentumhoz tartozó index-állományban, és a kiválasztási feltételnek megfelelő anyagok az itt felsorolt helyeken található dokumentumok összessége (logikai VAGY) lesz.

Ha tehát egyszer majd a számítógépes párkereső szolgálathoz fordulunk, hogy az oda beküldött életrajzok alapján a gép válassza ki azt a leendő feleséget (férjet), aki szép

is, jó is, gazdag is, okos is és pont olyan embert keres házastársnak, mint amilyenek mi vagyunk, akkor biztosak lehetünk benne, hogy a sok-sok jelentkező közül a gép az igazit (mormonoknál és mohamedánoknál az igaziakat) fogja megtalálni.

De ha mégsem bízunk ebben, akkor maradjunk a hagyományos párkeresési eljárásnál. Legföljebb amit megtakarítunk a gépi párválasztásért kifizetendő díjon, azt majd később fizetjük ki a válóperes ügyvédnek.

59.

A BÜROKRATA RÉMÁLMA
(Az automatizált iroda)

— Mancika, hol van az ALSZÁM-mal kötött szerződésünk, amelyben 20% engedményt kell adnunk, ha nem szállítjuk le határidőre, kifogástalan minőségben a húszezer kócficet? — szól ki a főnök a hangosbeszélőn a titkárnőjének.

Mancika leüt néhány gombot a mellette levő számítógép billentyűzetén, behívja a lemezről a dokumentum-visszakereső programot (könyvünkben ezt az 58. fejezetben találná meg). Megadja a gépnek a kérdésre jellemző szavakat (ALSZÁM, szállítás, 20% engedmény, kócfic), mire a főnök asztalán levő képernyőn két másodperc múlva megjelenik a szerződés teljes szövege.

A főnök kétszer is figyelmesen végigolvassa a szerződést.— Nagy baj van! — állapítja meg. — Ha nem változtatunk valamit sürgősen a termelési terven, akkor nem tudunk időben szállítani és fizetnünk kell. Méghozzá sokat. Mit lehetne tenni? Végig kell vizsgálni az összes reális lehetőséget. Mi lenne, ha . . .

De erre ott van a számítógép. Mert a főnök azért főnök, hogy már ehhez is értsen. Ahelyett, hogy a helyetteseit és szakértőit hívná, behívja a gépbe a termelési terv különböző módszereit elemző programot. Megváltoztat néhány adatot és a gép másodpercek alatt megadja szépen formázott táblázatban és grafikonokban, milyen következményekkel járna ezeknek a végrehajtása.

Ez még nem az igazi. De néhány gyors próbálkozás után a gép szolgáltat egy olyan változatot, ami lehetővé teszi a kócfic gyártással kapcsolatos határidő és minőség betartását úgy, hogy az más termékek előállításában csak olyan eltolódást okoz, melynek nincsenek hátrányos következményei.

Erről a döntésről értesíteni kell a gyár érintett részlegeit és a kócfic gyártásba bedolgozó alvállalkozókat, — dönt a főnök. Előveszi hordozható diktafonját, bemondja az érintettekre vonatkozó speciális adatokat, majd behívja Mancikát. — Címzés, bevezetés, udvariassági szöveg a szokásos, a lényeg a hangszalagon — nyújtja át eddigi munkája eredményét. — Kapják a kócfic gyártással és értékesítéssel foglalkozó belső részlegek és külső alvállalkozók.

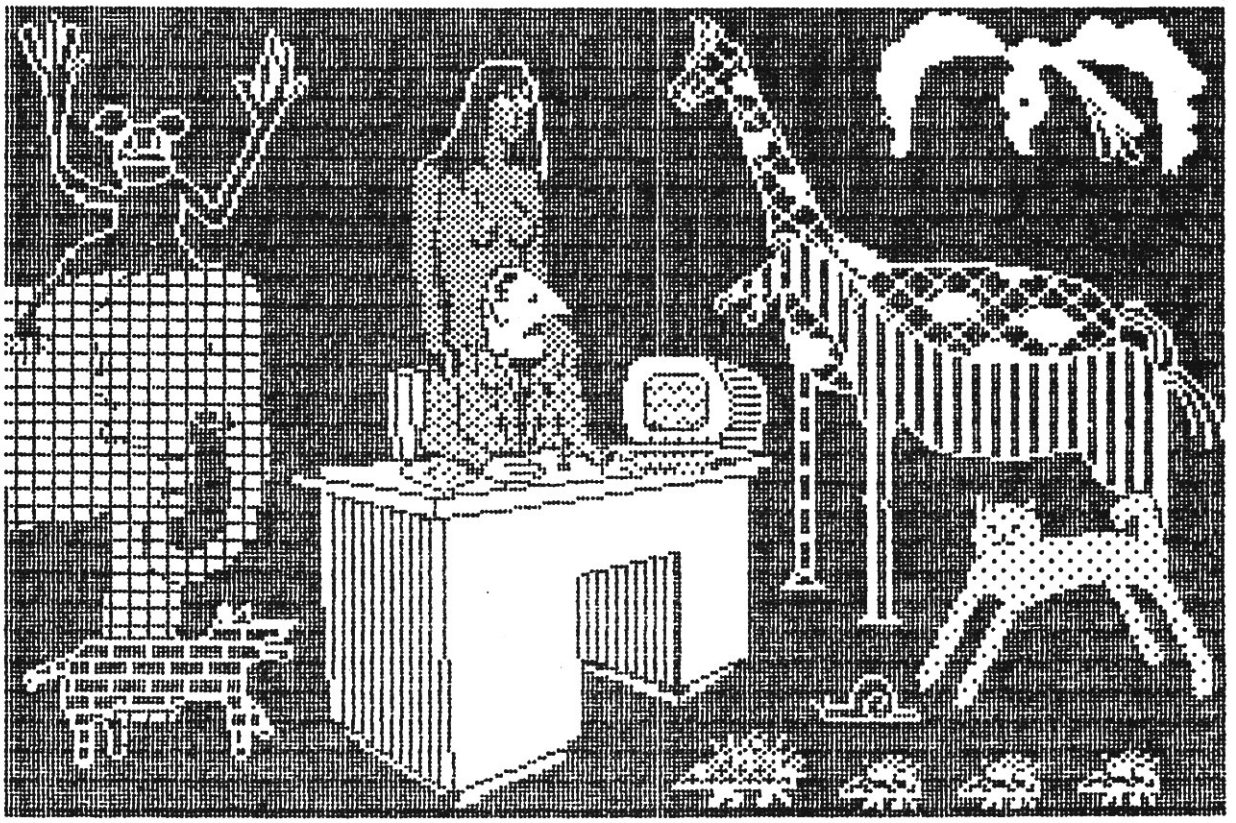
Mancika ismét a számítógéphez ül. Lekérdezi a kócficcel foglalkozók címlistáját, majd behívja a (könyvünk 55. fejezetében megismert) szövegszerkesztőt. Kijelöli a szabványos, mindenkinek szóló szövegrészeket, kiegészíti a személyekre szóló adatokkal és máris kinyomtatja a külső bedolgozóknak szóló leveleket. Hozzájuk postán, csak néhány nap múlva jut el az üzenet. A bentiek azonban azonnal megkapják, pedig nem is küldönc jár a hírrel.

A vállalat mindegyik számítógépe rá van kötve a helyi számítógépes hálózatra. A küldöncök helyett vezetékek viszik az egyes részlegek között az üzeneteket, kézbesítőkönyv helyett mágneslemezen iktatják a ki- és bemenő *„elektronikus postát"*. A hálózat minden gépének saját címe van és az üzenetet csak annak vagy azoknak a gépe fogadja, aki(k)nek az szól. A vételt a fogadógép nyugtázza, és az üzeneteket sorra elteszi a lemeznek erre a

célra kialakított területére, az elektronikus postaládába. Bekapcsoláskor, vagy egy megfelelő utasításra bármikor megjelennek azok az üzenetek, amelyek időközben a többi számítógéptől ideérkeztek. Azok, amelyek már érdektelenek (mi magunk, vagy más intézkedett ez ügyben, – vagy magától elintéződött) kitörölhetőek. Így a postaládánkban mindig csak az van benne, amivel még foglalkoznunk kell(ene).

„elektronikus posta"

Többször említettük már, hogy az emberi munka egyre inkább felértékelődött. Hosszú távon minél több emberi tevékenységet tudunk géppel kiváltani, annál gazdaságosabban tudunk dolgozni. Az irodai munka az egyik leginkább embert igénylő tevékenység. Ezt mutatja az is, hogy az adminisztrációs létszám az elmúlt ötven évben szinte mindenhol gyorsabban nőtt, mint a termelői. Ezen az irányzaton még a számítógépek sem tudnak változtatni.



Hiba azonban azt gondolnunk, hogy az irodai számítógépeknek fő célja a „papírmunka" megkönnyítése. Ha ez igaz lenne, akkor számítógépekkel automatikusan előállítható szabványlevelek tömegét küldenénk szerteszét, ahol azokat más számítógépek iktatnák, osztályoznák, de akiknek érdemben kellene foglalkozni vele, azok a rájuk szoruló információözön miatt kínjukban csak azt tudnák tenni, hogy a papírkosarak méretét mindenhol kétszeresére növelik. Az „emlékező" elektromos írógép, a nagy sebességű másológépek, a szövegeket, képeket nagy sebességgel továbbító telefaxok, az automatikus telefonhívók mind az adatok mennyiségi kezelését biztosítják, az információk minőségéhez, hasznosságához valóban nem sok közük van. Ha a főcél az irodai adminisztráció, a rutin jellegű ügyviteli munka automatizálása lenne, akkor a számítógép legjobb esetben a bürokraták produktivitását növelné meg. Rosszabb esetben azonban az információk tömegével elvonja a vezetők figyelmét a lényeges, a döntést igénylő problémáktól, a vállalat jelenét és jövőjét meghatározó kérdésektől.

Éppen ezért az iroda automatizálásának sem a berendezésektől kellene kiindulni és megkeresni az azokkal megoldható feladatokat, hanem fordítva. A vállalat szempontjá-

ból legfontosabb feladatok megoldásához kell megkeresni az ahhoz szükséges eszközöket. Ha tehát van egy számítógépünk és egy szövegszerkesztő programunk, abból még nem következik, hogy a gépet a levelezés lebonyolítására kell használnunk. Lehet, hogy ez a legkézenfekvőbb, de valószínűleg sokkal eredményesebben használnánk fel, ha olyan programokat futtatunk rajta, melyek a vállalat termelési, értékesítési stratégiájának kidolgozásához nyújtanak segítséget.

Persze, az igazi az, amikor erre is, arra is jut gépkapacitás. Ráadásul úgy, hogy ezek a gépek nemcsak egymással, hanem vállalati vagy területi országos nagygépekkel is kapcsolatban állnak, azok adataihoz is — ha jogosultak rá — hozzáférnek.

A bürokraták rémálma, hogy egyszer majd a számítógépek veszik át az uralmat az irodákban. Jogos-e vajon ez a félelmük? Valóban helyettesíteni tudják-e a gépek az ügyfeleket és az iratokat ide-oda küldözgető hivatalnokokat?

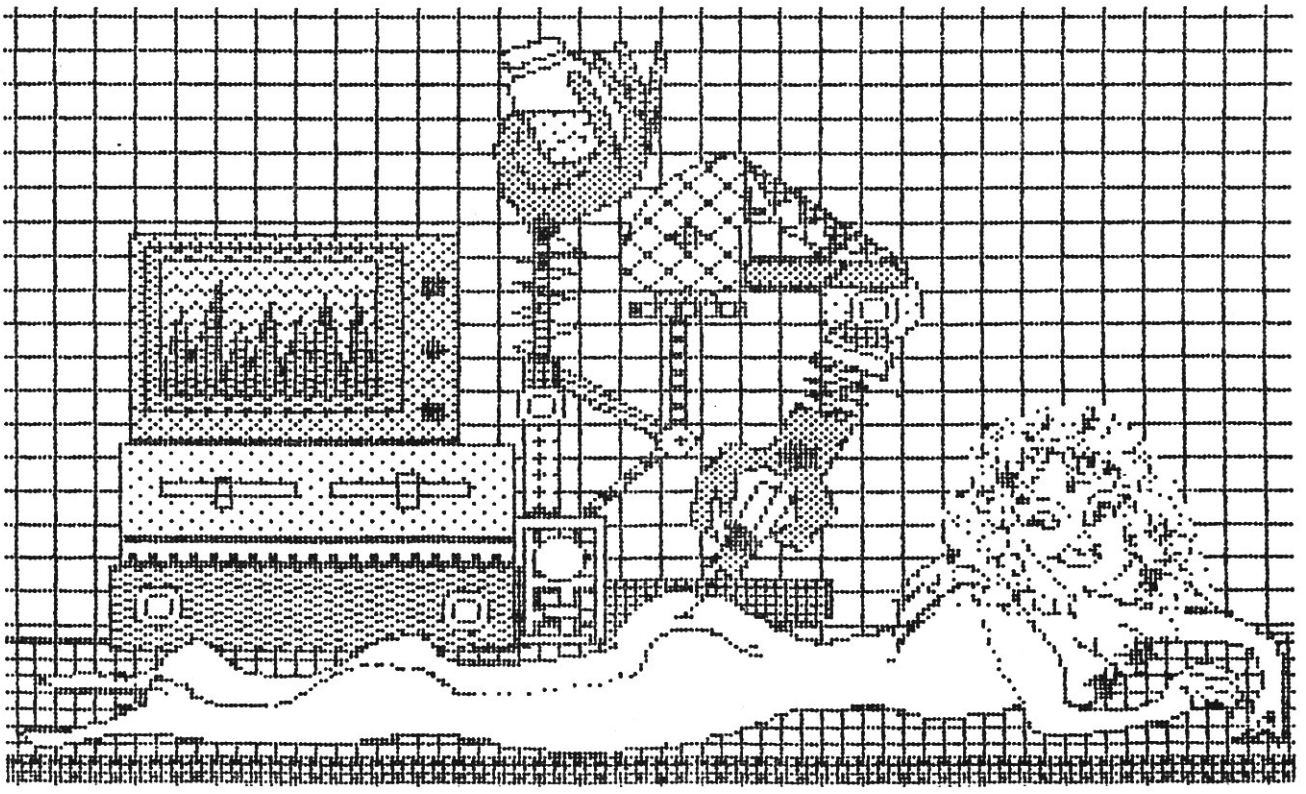
Az irodai gépek csodálatosak. A munkába való bevonásuk öröm és áldás. De minden teljesítőképességük, tárolókapacitásuk, sebességük ellenére sem tökéletesek. Hatékonyságuk a géptípustól és az alkalmazott szoftvertől függően igen eltérő. Gyakran érthetetlen módon viselkednek, nem értik meg, mit akarunk tőlük. Sokszor lustán, lassan válaszolnak, sőt időnként, minden látható ok nélkül részben vagy teljesen működésképtelenné válnak. Ha nem látjuk el őket munkával, órákig is képesek tétlenek maradni. Az egyik nap még tökéletesen működnek, a következő napon pedig már a legegyszerűbb feladatot sem tudják megoldani. Így tehát valóban megvan minden reményünk, hogy egyszer majd tökéletesen helyettesíteni fogják az emberi lényeket, hiszen — a fentiekből nyilvánvaló — munkaszokásaik legfontosabb jellemzőit már most is átvették.

60.

DR. (MED) UNIV. SZÁMÍTÓGÉP
(Számítógépes diagnosztizálás)

A betegek diagnosztizálása és kezelése az egészségügyi ellátás fő funkciója. Az orvostudomány mai állása szerint kb. 30 ezer betegségből kell az orvosnak kiválasztani azt az egyet vagy egynéhányat, amely a vizsgált betegnél megtalálható. A klinikai döntés legsarkalatosabb pontja, hogy az orvosnak nagyon sok adatot kell értékelnie. A kórelőzmény adatai, a tünetek és jelek, a rutinszerűen kivitelezett laboratóriumi és egyéb vizsgálatok eredményei által meghatározott számos lehetőség közül kell választania és kialakítania a beteg állapotára vonatkozó hipotéziseket. Sok száz kórtörténet kiértékelése alapján kiderült, hogy a tünetek száma egy-egy betegségnél a 80—100-ast is elérheti.

Világszerte folynak a kísérletek azzal a célzattal, hogy a számítógép lehetőségeit a diagnosztizálás, a gyógykezelés területein minél hatásosabban tudja az orvos közvetlenül a beteg érdekében hasznosítani. Az orvosi döntéshozatal számítógépes támogatásának területén az utóbbi években több, egymástól alapvetően eltérő kezdeményezés bontakozott ki. Ezek közül mutatunk be néhányat.



Az egyik módszer az orvosi döntést úgy segíti, hogy a vizsgált betegnél megállapított betegségtünetek alapján korlátozott számú, ismert, jól definiált betegség közül választja ki az objektíve legvalószínűbb diagnózist. A módszer feltételezi, hogy a tünetek valószínűsége, a betegségek gyakorisága ismert, hogy a tünet-valószínűségek függetlenek egymástól, valamint azt, hogy az adott beteg csak a számítógépben nyilvántartott betegségek egyikében szenved.

A fenti módszerhez igen közel áll az a programrendszer, amely, mint megbízható konzultáns segíti az orvost bármilyen diagnosztikai vagy terápiai döntésben, a diagnosztikai következtetések folyamatát és a terápia stratégiák értékelését becslési, számítási feladattá redukálva. Az ismert tünetek, vizsgálati eredmények és a kórelőzmények alapján a program kiválasztja a szóba jövő betegségeket, és valószínűségi megoszlásuk alapján rangsorolja őket, javaslatot tesz esetleges további vizsgálatokra, ill. a megfelelő gyógykezelésre.

Egy harmadik módszer úgy igyekszik segíteni az orvos önálló diagnosztikai munkáját, hogy az adott betegségre vonatkozó jellemző kórképet bocsátja az orvos rendelkezésére. Ez a módszer főként olyan betegségek esetén hatékony, amelyek tüneteit az orvos nem ismeri feltétlenül.

MEDIAC

A gyakorlatban is megvalósult alkalmazások közül a *MEDIAC* elnevezésű orvosdiagnosztikai számítógépes rendszer működését mutatjuk be.

A rendszer egy központi számítógépen alapul. Az orvos munkahelyéről, telefonon kérhet speciális orvosi szempontok szerinti diagnózist.

A megfelelő telefonszámot tárcsázva az orvos közvetlenül összeköttetésbe kerül a géppel. A beteg tüneteit, a vizsgálati eredményeket és az összes egyéb információt a telefontárcsa segítségével, szám- és betűkódok formájában továbbítja a géphez. A telefon fülhallgatóján keresztül a gép hallható beszéd formájában visszaolvassa a betárcsázott adatokat, így az orvos ellenőrizheti, hogy a tárcsa útján közölt adatokat valóban jól küldte-e el a számítógépnek. Bármely tévedését az orvos újratárcsázással korrigálhatja. Az eredmény néhány valószínűsíthető diagnózis, azok valószínűségi sorrendjében, és egyben további kiegészítő vizsgálatok ajánlása. Ezeket az eredményeket 30 másodpercen belül kapja meg az orvos, mégpedig élőszóban, a telefon fülhallgatóján át. A szöveg ismétlése mindaddig tart, amíg az orvos le nem teszi a telefont.

Az orvos és a gép között egy speciális beviteli-kiviteli egység közvetít. Feladata, hogy az orvos tárcsázott üzenetét a gép nyelvére kódolja, ill. a gép válaszait hangokká alakítsa. A feldolgozó egység értékeli a beteg tüneteit, és ezek alapján adja meg a választ.

Az orvos a telefontárcsával először egy speciális jelet tárcsáz, amellyel jelzi, hogy megkezdi a tünetek közlését. Minden tünetet háromjegyű kód képvisel, és egy újabb speciális kódjel tárcsázandó a befejezés után. A gép 128 tünetet képes kezelni.

A feldolgozó egység összehasonlítja az adott, betáplált tünetegyüttest mind a 2560 általa tárolt tünetprofillal és kiválasztja az öt legvalószínűbb betegséget, azon az alapon, hogy azok tünetprofilja mennyire hasonlít az általa tároltakhoz, egyúttal számításba veszi az egyes tünetcsoportok mérlegelésével kapcsolatos szempontokat is.

A kimenő üzenet úgy fogalmazódik meg, hogy az előre, végtelenített magnószalagra felvett szavak közül a megfelelőeket a gép kiválasztja és a kívánt sorrendben átjátssza azokat egy kiviteli magnószalagra. Ezt követően a szalagon levő szöveget mindaddig ismétli a telefonba, amíg a kagylót le nem teszik.

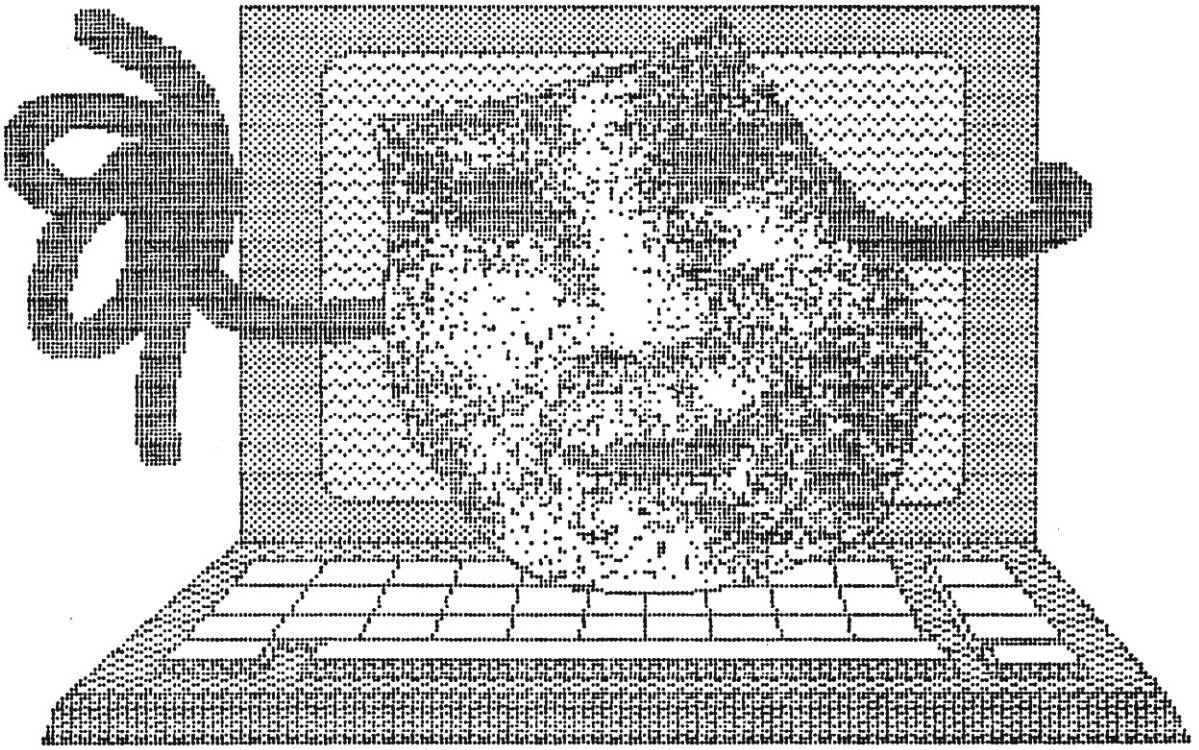
A számítógép tárkapacitása lehetővé teszi, hogy az ismert betegségek összes jellemzőikkel együtt tárolhatók legyenek, és egy konkrét esetben a beteg problémáit értékelve, a szóba jöhető betegségek mindegyikét az orvos rendelkezésére bocsássa.

A számítógépes és a hagyományos diagnosztika alapjaikban különböznek egymástól. Ha a számítógép helyet is kap a klinikai diagnosztikában, akkor sem helyettesítheti az orvost, mindössze segítheti a gyógyító munkát. Egyes jövőkutatók véleménye szerint a számítógép belátható időn belül az orvosi munkának ugyanolyan megszokott segédeszközévé válik, mint manapság a sztetoszkóp, bár nem valószínű, hogy az előbbit is a nyakukban fogják hordani a doktor urak.

61.

SZIMULÁL(J)UNK!
(Kiképzés számítógép segítségével)

Nem tudjuk, hogy az előző fejezetben szereplő számítógép milyen diagnózist adna egy szimuláns esetében. Ennek a fejezetnek a címszava viszont mást jelent, és talán kissé pontatlan is, mert ezt kellett volna írni: szimulációval foglalkoz(z)unk. De ebben az esetben félő, hogy sokan el sem olvasnák e fejezetet. Pedig a szimuláció az egyik legizgalmasabb területe a számítógépek alkalmazásának. Miről is van szó?



A valós világ létező rendszereinek tanulmányozása sok esetben nem lehetséges közvetlen módon, mert például veszélyes, túlságosan drága, vagy megoldhatatlan. Gondoljuk csak el, milyen veszélyeket rejt egy olyan kísérlet, amely atomerőműben zajló folyamatokat kíván vizsgálni szélsőséges körülmények között egy működő rendszerben. Ugyanez igaz azoknál a rendszereknél is, amelyek még csak a tervezés fázisában vannak. Mekkora felesleges költségekkel járna például megépíteni egy új konstrukciójú repülőgépet, amelyről a próbaúton derülne ki, hogy a hangsebesség átlépésekor letörnek a szárnyai.

számítógépes szimuláció, modell

Beláthatatlanok lennének a következményei egy olyan kísérletnek is, amely a Balaton ökológiai egyensúlyát változtatná és figyelné, hogy mindez milyen hatással van a tóra.

Az ilyen és ehhez hasonló jellegű problémák megoldásának egyik módja a *számítógépes szimuláció* alkalmazása. A szimulációs modellezés lényege, hogy matematikai és számítástechnikai módszerekkel létrehozunk, leírunk egy olyan rendszert, amely az eredeti (valós) rendszert a vizsgálat szempontjából megfelelő mértékben és módon helyettesíti. Azt a rendszert, amely az eredetit a vizsgálat során helyettesíti, *modellnek* nevezzük.

Mivel az eredeti rendszerek nem statikusak, vagyis az idővel (és a külső tényezők hatására) változnak, ezért a modellnek is így kell viselkednie. Az időbeli változások a matematikában differenciálegyenletek segítségével írhatók le. Egy modell viselkedése leírható azonban az elkülönülő időpontokban bekövetkezett változások segítségével is. (Ez utóbbi hasonlít a filmkészítéshez: állóképek egymásutánja adja a mozgás képzetét.) A szimulációs modellek általában analitikus módon (a differenciálegyenletek megoldásával) nem, vagy csak nagyon bonyolult módon kezelhetők, ezért kap nagy szerepet a számítógép. Ennek segítségével (az egyes állapotok leírásával) numerikus módon oldhatók meg ezek a modellek.

repülés-szimulátor

A számítógépes szimuláció egyik leglátványosabb alkalmazási területe a pilóták képzésére és „edzésére" szolgáló *repülésszimulátor.* Ezek közül a Lufthansa Légitársaság szimulátoráról ismertetünk néhány érdekes mozzanatot.

A szimulátorok használata során fontos, hogy a pilóta (vagy leendő pilóta) teljes mértékig úgy érezze, hogy valódi repülőgépen ül. Ez nemcsak azt jelenti, hogy a pilótakabint élethűen kell kialakítani, hanem azt is, hogy gondoskodni kell a zaj-, a kép-, a mozgás szimulálásáról. A repülésszimulátorban a pilótafülke „modellje" minden tekintetben megegyezik az eredetivel. A műszerek elhelyezkedése, a pedálok, kézi vezérlők elrendezése mind-mind az eredeti hű mása.

A pilóta látómezőjén kívül helyezkedik el az oktató ülése. Erről a helyről tudja az oktató vezérelni a „gépre" ható külső tényezőket (nyomás, hőmérséklet, szélirány és sebesség, látási viszonyok stb.), valamint a belső tényezőket (raksúly, üzemanyag, súlypont elhelyezkedés stb.). Az oktató kb. 300 féle fedélzeti hibát tud előállítani részben a billentyűzetről történő, részben grafikus adatbevitellel.

A szimulátort háromféle feladat végrehajtatására alkalmazzák:

- átmeneti tréningre, vagyis ha egy pilóta egy újabb típuson kezd repülni, itt készül fel,
- félévenkénti vizsgára,
- félévenként ismeretfrissítő és gyakorlati képességeket vizsgáló tesztelésre.

A szimulátor fontos szolgáltatása, hogy az elmúlt 5–10 perc valamennyi szimulátoradata visszaállítható, s ily módon elemezhető. Ez lehetővé teszi a tréningszerű gyakorláson kívül, hogy a pilóta az elkövetett hibákat „kívülállóként" újra megtekinthesse.

A szimulátor lelke az a számítógép, amely minden információt feldolgoz és ezekre válaszol. Erre a célra akármilyen számítógép nem alkalmas. Az ismertetett rendszert kiszolgáló gép néhány jellemző adata: 6 Mbájt központi tár, 80–100 Mbájt háttértár, 0,9–2 MOPS (millió művelet/s) sebességű processzor.

A jó hegymászó az öreg hegymászó – tartja egy mondás. A pilóták is nyilván ezt vallják, még akkor is, ha a szimulátorban több órát repülnek, mint a valóságban.

62.

MIKOR ESIK LE A TANTUSZ?
(Elektronikus pénz)

Feljegyzés a jelenkor számára: tantusz = pénzt helyettesítő érme, pl. utcai telefonok használatához, Postán, dohányboltban árusították, 60 fillérbe került.

Ma már tantusz nélkül telefonálunk. Az utcai telefonok pénzérmékkel működnek. Vajon elképzelhető-e, hogy egyszer majd pénzérmék nélkül használhatjuk ugyanezeket a telefonokat? Elképzelhető-e pénz nélküli pénz?

Aki úgy gondolja, hogy ez képtelenség, gondoljon egy pillanatra a csekkre, vagy a hitelkártyára. Ezek olyan eszközök, amelyek már ma is készpénz kímélő, azt helyettesítő funkciókat látnak el. Nagy előnyük, hogy nem kell pénzt hordani magunkkal, rettegve attól, hogy ellopják, vagy elveszítjük. Persze a hitelkártya elvesztése is okozhat problémákat, sőt maga az elvesztett csekk is adhat lehetőséget a visszaélésre.

Bizonyára mindezt végiggondolta Roland Moreno francia mérnök is, aki 1974-ben szabadalmaztatta ötletét, a *memóriakártyát.* Ez a kártya kinézetre olyan, mint a hagyományos hitelkártya, vagyis plasztik alapanyagú, mérete 85.6 mm X 54 mm X 0.76 mm, de egy mikroprocesszor és egy (esetleg több) tárolóelem is bele van építve. A mikroprocesszor és a tár egymással, és a környezettel való kapcsolatot biztosító érintkezőkkel össze vannak kötve.

aktív memória-kártya

A beépített mikroprocesszor a kártyát *aktívvá* teszi, vagyis a kártya képes a külvilággal tartott kapcsolat során döntéshozatalra. Ilyen döntés lehet például adott esetben a kártyában tárolt információk titkosságának megőrzése, az illetéktelen hozzáférési kísérletek meghiúsítása.

A memóriakártya programozható, tárolókapacitása 1–16 kbit lehet. A kártya tulajdonosának azonosítását biztosító adatokon kívül így más adatok is tárolhatók benne. A mikroprocesszorban levő program megfelelő biztonsággal védi a titkosnak szánt adatokat az illetéktelen hozzáféréstől.

Az aktív memóriakártya sok területen felhasználható, így „elektronikus pénztárcaként" is. Kétféle változata lehetséges. Az egyszerűbb és éppen ezért olcsóbb az egyszer feltölthető kártya, amely nyilvános telefonok, fizető parkolóhelyek igénybevételét teszi lehetővé. Ha a kártya PROM-jába egyszer bevitt „pénzösszeget" a tulajdonos elhasználta, a kártyát ki kell cserélni.

A bonyolultabb változat igazi „pénztárcaként" tud működni, nemcsak kivenni, hanem beletenni is lehet „pénzt". Az ilyen kártyák használatához három „résztvevő" szükséges. Az egyik a bank, vagy pénzintézet, amely az ügyfélnek kiadja a memóriakártyát. A banknak ehhez rendelkeznie kell olyan eszközzel, amely az üres kártyát feltölti: beégeti a memória védett részébe azt a programot, amely a használat során a megadott titkos kód hatására hozzáférést enged a memória pénzösszegeket tartalmazó részéhez. (Ezzel válik lehetővé, hogy a vásárlások során rögzíteni lehessen a kártyán a felhasznált összegeket és a vásárlás körülményeit.) Ugyancsak a banki eszköznek kell tudnia feltölteni a memóriának azt a helyét, ahová az elkölthető pénzösszeg kerül beírásra. Ezen kívül a kártyába be kell írni néhány technikai adatot: a bank nevét és azonosító kódját, a kártya érvényességi

idejét, az ügyfél bankszámla számát stb. Ugyanakkor a kártya felületén (látható módon) is meg kell jeleníteni az ügyfél nevét, a kártya azonosító sorszámát, a bank azonosító kódját. Lehetőség van rögzíteni pl. az ügyfél aláírását, sőt fényképét is. A kártya kiállításakor természetesen a bank központi számítógépében is rögzítik az ügyfél nyilvántartásához szükséges adatokat A kártya újraérvényesítése és feltöltése hasonló módon történik.

Az alkalmazásban résztvevő másik fél az az áruház, illetve azok a helyek, ahol a kártyát használni lehet. Ez a használat (vagyis maga a vásárlás) speciális pénztárgépeket igényel. A pénztárgép lehet on-line, vagyis közvetlenül az áruház központi számítógépére kapcsolt egység, és lehet off-line, azaz önálló berendezés. Ez utóbbi esetben a számlázási adatokat mágneses adathordozón (kazettán vagy hajlékony lemezen) gyűjtik, és csak műszak után juttatják el a számítógéphez. A mágneskártyát kezelő pénztárgépnek viszont a blokkon és a saját számítógép számára készített gyűjtésen kívül kezelnie kell a memóriakártyát is, amelybe szintén be kell írnia, hogy hol, milyen értékű vásárlás (pénzfelhasználás) történt. Ez csak akkor lehetséges, ha a pénztárgépnek van memóriakártya olvasója.

Az alkalmazás harmadik résztvevője maga a vásárló, akinek a memóriakártya használatakor egy erre a célra szolgáló eszközön meg kell adnia a saját titkos kódját. Ez akadályozza meg végképp a memóriakártyákkal történő visszaéléseket. (Ez a szerkezet az eladónál levő pénztárgépbe van beépítve.) Az eladó a kártya azonosítási adatai alapján ellenőrizheti, hogy a szóban forgó kártya nem szerepel-e feketelistán (lopott kártyáról, esetleg más okból letiltott kártyáról van szó).

Láthatjuk tehát, hogy viszonylag egyszerű, biztonságos ügymenetről van szó. Ennek megvalósítása viszont igen beruházásigényes. Gondoljuk csak meg, hogy a memóriakártyák előállításán kívül mind a bankokat, mind a potenciális eladóhelyeket fel kell szerelni cseppet sem olcsó berendezésekkel. De ha ezeket az egyszeri ráfordítási költségeket összevetjük azokkal a költségekkel, amelyek a hagyományos pénzforgalom mellett felmerülnek (pl. mint legnagyobb tétel az elhasznált pénzek állandó pótlása), kiderül, hogy hosszú távon a memóriakártya az olcsóbb. Ráadásul használata biztonságosabb, kényelmesebb, gyorsabb, kevesebb adminisztrációval jár, mint a hagyományos készpénzforgalom. (Bécsben már ma is utcai automatabankok segítségével juthat a memóriakártya tulajdonosa egyszerűen és gyorsan készpénzhez, Franciaországban pedig a legtöbb utcai telefonfülkéből már csak memóriakártyával lehet telefonálni).

Nem véletlen, hogy Magyarországon 1984-ben megalakult az Aktív Memóriakártya Gazdasági Társaság, amelynek 25 tagja között pénzintézetek, kereskedő cégek, szervezőkutatóintézetek és gyártó vállalatok találhatók. Ez azt jelenti, hogy idehaza is leesett a tantusz. Egyelőre kísérleti jelleggel egyik kisvárosunkban megteremtik az elektronikus pénzhasználat feltételeit. Most már csak az a kérdés, mikor lesz aktuális ez a feljegyzés az utókor számára: pénzérme és bankjegy = régi fizetési eszköz, amelyből sohasem volt elég, mégis gyakran elvesztette a tulajdonosa.

63.

TUVUDSZ IVÍGY BEVESZÉVÉLNIVI?
(Rejtjelezés számítógéppel)

A különböző helyeken levő számítógépek és terminálok között az információ az adatátviteli vonalakon elektromos impulzusok formájában áramlik. Ezeket, akárcsak a telefonbeszélgetéseket, le lehet hallgatni, különlegesen érzékeny lehallgató készülékkel akár nagyobb távolságból is. Ily módon a technikailag kellőképpen felkészült adattolvaj mindazon információkhoz hozzájuthat, melyek a felhasználók számára bizalmas adatokat szállítanak. Az adatátviteli vonalaknak az ilyen típusú lehallgatástól való leárnyékolása – már a jelenlegi adatforgalom mellett is – anyagi okokból megvalósíthatatlan lenne. Ennek ellenére az adathálózaton keresztül áramló adatok titkossága gyakorlatilag 100 százalékosan biztosítható megfelelő matematikai eljárásokkal. A már szabványosnak tekinthető módszer segítségével a szöveg kódolását illetve dekódolását a számítógép másodpercek vagy legföljebb néhány perc alatt elvégzi a megfelelő kulcs ismeretében, míg annak hiányában még a leggyorsabb gépnek is évbilliókig tartana, amíg a szokásos 2^{55} féle megfejtési lehetőség mindegyikét végigpróbálnál.

A titkosítás elve – némileg leegyszerűsítve – a következő. A titkosítandó szöveget adott hosszúságú, példánkban 4 jelből álló csoportokra bontjuk. Ezekben mindegyik jelet a csoportokban elfoglalt helyétől és a *kódolási kulcstól* függően egy másik jellel helyettesítünk. Például, ha a kulcs erre vonatkozó része 1420, akkor minden csoportban az első jel helyett az ABC-ben az 1-gyel, a második jel helyett a 4-gyel, a harmadik helyett a 2-vel utána következőt vesszük, míg a csoport negyedik, utolsó jele változatlan marad. Így az A betűből, attól függően, hogy egy négyes csoporton belül hol helyezkedett el, B, E, C lesz, vagy A marad. Ezáltal példánkban – csupán betűkből álló szöveggel foglalkozva – ha a hosszú és ékezetes magánhangzóktól, valamint a kettős mássalhangzóktól eltekintünk és a Z után téve a szóközt, majd újra kezdve az ABC-t – az

kódolási kulcs

„EZ ITT TITOK"

szövegből az

„EZ I", „TT T", „ITOK" csoportokat,

majd az „FCBI", „UXBT", „JXQK" jelsorozatokat kapjuk. (E-ből F, Z-ből C, a szóközből B lesz, az I megmarad, és így tovább.)

Ezután ezen csoportok betűit felcseréljük egy másik, például a 3142 kulcs szerint. Ez azt jelenti, hogy minden csoportban a 3. jel kerül az első, az 1. a második, a 4. jel a harmadik és 2. a negyedik helyre. Így a „BFIC", „BUTX", „OJKX" kétszeresen kódolt jelsorozatokhoz jutunk. Ha most a kulcsot kiegészítjük 14-gyel, ami azt jelenti, hogy minden csoportot két találomra kiválasztott jellel kibővítünk, melyeket az első és a negyedik helyre szúrunk be, akkor végül a

"PBFXICSBUJTXAQJNKX"

végső kódolt szöveghez jutunk. (A beszúrt jeleket a szemléltetés kedvéért vastagabban szedtük.) Aki nem ismeri a kulcsot, annak számára ez teljesen megfejthetetlen, még akkor

is, ha pontosan ismeri a kódolási eljárást. Aki viszont tudja, hogy a teljes kulcs értéke 1420314214, az előbb 4+2=6-os csoportokra osztja a kódolt szöveget, elhagyja a töltelékként beszúrt első és negyedik jelet minden hatos csoportból, visszaállítja az eredeti sorrendet a 3142 kulcs szerint (az első jelet a 3. helyre, a másodikat az 1., a negyedik jelet a 2., a harmadikat a 4. helyre rakra a csoportokban a példában jelzett nyilak szerint), majd az 1420 kulcs alapján visszaszámolja az ABC-ben minden egyes betűre a megfelelő karaktert. Ezáltal dekódolta, visszaállította az eredeti szöveget. Tetszés szerinti üzenetek ilyen módon való visszaállítására, illetve megfejtésére kész programok vannak. A géppel csak a kulcsot kell közölni. Gondolni kell arra is, hogy esetleg a kulcs kerül illetéktelenek birtokába. Éppen ezért időnként a kulcsot is változtatják, szükség esetén akár naponként vagy üzenetenként is. Ilyen esetekben legelső üzenetként az éppen aktuálissá váló kulcsok mennek keresztül az adatátviteli vonalon, természetesen ugyanilyen módon titkosítva. A kulcsok dekódolására szolgáló kulcs, az úgynevezett vezérkulcs állandó. Értékét legföljebb egy vagy néhány, teljes mértékben megbízható személy ismeri. De az is lehet, hogy külön-külön csak egyes részeit tudják és csak együttesen képesek a rejtjelzett kulcsokat hozó üzenet megfejtésére és a további üzenetek dekódolására szolgáló új kulcs(ok) megmeghatározására.

Hasonló módon lehet titkosítani a számítóközpontban levő adathordozókon tárolt bizalmas adatokat. Így ezeket hiába lopnák el vagy készítenének róla másolatot, tartalmukat csak a kódolási kulcs birtokában lehet megfejteni. Hej, ha szegény Sándor Mátyásnak lett volna számítógépe!!

64.

A PÉNZ BESZÉL – A GÉP DOLGOZIK
(A számítógépes feldolgozások költségei)

A számítástechnikának az elmúlt évtizedben robbanásszerűen bekövetkezett terjedésében jelentős szerepet játszott a gépek teljesítményének növekedése, használatuk egyszerűbbödése, méretük csökkenése. Mindezek a tényezők azonban csak azért tudtak ekkora hatást kifejteni, mert a számítástechnika *gazdaságossá* vált. Az alkatrészek szabványosítása, nagy tömegben való előállítása lehetővé tette, hogy az otthoni „játék" számítógépek ára alacsonyabb, mint egy televíziókészüléké és a kisebb teljesítményű professzionális személyi számítógépek ára összemérhetővé vált az irodai munka automatizálását elősegítő egyéb berendezésekével.

Ahhoz, hogy a számítástechnika gazdaságosságát érzékelni és érzékeltetni tudjuk, számokra van szükségünk. Pontos adatokat azonban igen nehéz adnunk, mivel egészen mások az árarányok az iparilag fejlett tőkés országokban, mint hazánkban és ezek az árak is folyamatosan változnak, a hardvert tekintve állandóan csökkennek. Csak néhány példa: Egy 2 Mbájt központi tárú, 2–300 Mbájtnyi lemeztárolóval kiegészített IBM nagyszámítógép a hetvenes évek elején a szükséges perifériákkal (vezérlő konzol, kártyaolvasó, sornyomtató, mágnesszalagos egységek) 1–1,5 millió dollárba került. Ma egy ugyanilyen kiépítettségű, jóval korszerűbb technológiával készült gép 250–300 ezer dollárért megkapható. Magyarországon 1985 végén egy 256 kilobájtos IBM PC/XT professzionális számítógépet egy beépített hajlékonylemez meghajtóval 1 millió forint körül árultak – és a vevő hálás lehetett, ha kapott. 1987-ben ugyanilyen teljesítményű távolkeleti gép már kb. 100 ezer forintért is megvásárolható – de ezért az árért sajnos most sincs mindig.

A konkrét, de gyorsan változó számok helyett inkább arányokon kívánjuk bemutatni a gazdaságosságot. A hatalmas méretű áresés azért érzékelteti, hogy sok helyen, ahol már korábban is alkalmazni lehetett volna a számítástechnikát, de nem volt még kifizetődő, ma már a gazdaságossági szempontok is kikényszerítik a számítógép használatát. Különösen érvényesül ez a fejlett ipari országokban, ahol az emberi munka ára igen magas.

A mérnöki munkát nagymértékben meggyorsítják az erre a célra szolgáló számítógépes mérnöki tervezési rendszerek. Egy ilyen egyszerűbb célra alkalmas PC ára a programokkal együtt 10 ezer dollár körül van, ami megfelel ott egy jó mérnök négy havi fizetésének. Így ha a számítógép segítségével a mérnök csak 30%-kal dolgozik hatékonyabban, akkor is egy év alatt megtérül a beruházás. Ilyen gyors visszatérüléssel pedig már nagyon is megéri a pénz befektetése. Hasonlóan gyors megtérülést tapasztalhatunk az ügyviteli munka számítógépesítésénél is.

Sajnos Magyarországon az árarányok nem ilyen kedvezőek. Nálunk egy hasonló számítógépet egy jó mérnök 10–15 évi fizetéséből lehetne csak beszerezni. Ez is egyik oka annak, hogy a számítástechnika a kelleténél talán még mindig lassabban vonul be mindennapi életünkbe.

A számítástechnika alkalmazásának költségébe mindenképpen be kell számolnunk a hardver árát. Személyi számítógépeknél ez általában a teljes vételár. Nagyszámítógépeknél azonban, amelyeken egyidejűleg egyszerre sokan is dolgozhatnak, gyakran célszerűbb

a gépet, illetve annak szükséges részeit bérelni, akár úgy, hogy a kívánt időre az egész gépet lefoglaljuk, akár úgy, hogy másokkal osztozva csak a ténylegesen igénybe vett erőforrásokért fizetünk. A gyakorlatban általában az utóbbi szokott előfordulni. (Ez az az üzlet, amelyikben mindkét fél jól jár. A vevő olcsóbban jut ugyanahhoz a szolgáltatáshoz, az üzemeltető pedig magasabb bevételt ér el, mert a multiprogramozás megvalósításával – lásd a 33. fejezetet! – ugyanabban az időben egyszerre több helyre adja el a gépet, anélkül, hogy ezáltal a törvény által büntetendő tisztességtelen haszonhoz jutna.) Ekkor az operációs rendszernek egy speciális programja figyeli, hogy az egyes felhasználók milyen erőforrásokat, mennyi ideig és milyen sürgősséggel vettek igénybe. Ugyanezért a szolgáltatásért többet kell fizetni, ha a munkának, a programnak magasabb volt a prioritása (lásd a 34. fejezetet!), azaz ütemezéskor a többi, ugyancsak jelen levő egyéb munkák elé sorolta a vezérlő program.

gazdaságos géphasználat

Minden erőforrásnak megvan az adott számítógép konfiguráción a díjszabása. A *géphasználatért* fizetendő összeg az alábbi tényezőkből tevődhet össze:

- a feldolgozáshoz felhasznált tiszta processzoridő,
- a feldolgozás alatt lefoglalt központi tár nagyságának és a felhasznált időnek a szorzata,
- a perifériákhoz való hozzáfordulások száma,
- a perifériák használatának összideje,
- a kiírt és beolvasott sorok száma,
- a terminál(ok) bekapcsolási időtartama,
- a feldolgozáshoz felhasznált speciális programtermékek használati díja,
- átalánydíj az operációs rendszernek a programok elindításánál, futásánál és befejezésénél nyújtott szolgáltatásokra.

Az egyes összetevőket úgy határozzák meg, hogy egy magyarországi nagyszámítógépen óránként 20–30 eFt legyen a géphasználati díjból származó összbevétel. Ezt természetesen nem egyetlen egy felhasználó fizeti ki, osztozik rajta az összes résztvevő, aki akkor a gépen dolgozott, a ténylegesen igénybe vett szolgáltatások arányában. Ez az összeg fedezi a gép amortizációját, az üzemeltetés anyagi és energiaköltségeit, az üzemeltető személyzet munkabérét és az üzemeltetéshez felhasznált szoftver bérleti díját vagy vételárának időarányos részét.

Ezek az üzemeltetési költségek a professzionális személyi számítógépek néhány száz forintos óránkénti bérleti díjához képest iszonyatosan magasak. De vegyük figyelembe, hogy egy ilyen nagyszámítógépen esetleg 30–40 ember is dolgozik egyidejűleg, és az egyes programok futásának átlagos ideje még így is nagyságrenddel rövidebb a nagygépen mint a PC-ken. Régebben azt mondták, hogy a számítógépek ára és teljesítménye között négyzetes összefüggés áll fenn. Egy kétszer olyan drága gép teljesítménye négyszer akkora, azaz ugyanannak a feladatnak az elvégzése feleannyiba kerül. Ez ma már nem érvényes ilyen mértékben, de most is fennáll, hogy minél nagyobb egy számítógép, annál gazdaságosabb vele a feldolgozás. Természetesen ez csak akkor igaz, ha a gépet folyamatosan el tudjuk látni munkával, hiszen amikor nem dolgozik, akkor sok pénzt érő berendezések állnak kihasználatlanul. Ezért van az, hogy ahol sok nagy feladatot kell megoldani, továbbra is hasznosnak bizonyultak a nagyszámítógépek, míg az eseti, viszonylag kisebb munkák ellátására ma már technikailag alkalmasak és gyakorlatilag kifizetődőbbek a személyi számítógépek. Ezért a különböző kategóriákba tartozó gépek továbbra is kiegészítik egymást, kicsire is és nagyra is egyaránt szükség van. De nem biztos, hogy sok kicsi sokra megy. Egy bizonyos feladatméreten felül még a személyi számítógépek összekapcsolásából adódó hálózat sem tudja felvenni egy nagyszámítógépes rendszerrel a versenyt sem teljesítőképességben, sem árban.

költség-tényezők

A rendszeres feldolgozás számítógépes költségein kívül még egyéb *költségtényezők*kel is számolnunk kell. Ezek egy része a rendszer kifejlesztésével kapcsolatos, másik része a rendszeres feldolgozáshoz szükséges adatok összegyűjtésének, gépre való bevitelének költségeiből adódik.

Bár a rendszerfejlesztés költségei meglehetősen magasak, a gazdasági-üzleti életben, ahol az egyszer létrehozott alkalmazási rendszerek hosszú ideig futnak változtatás nélkül vagy csak minimális változtatással a rendszeres üzemeltetésnek a rendszer élettartamára összegezett költségei összemérhetők, sőt általában magasabbak, mint a fejlesztésé és programkarbantartásé. Különösen igaz az, ha nem egyedi célra készült programokkal dolgozunk, hanem készen vásárolttal. Ezek ára lényegesen alacsonyabb, hiszen a fejlesztés költségei szétoszlanak az összes vásárlók között.

adatbevitel

Ügyviteli rendszereknél jelentős költségtényező lehet az *adatbevitel.* Az eredeti adatokat közvetlenül olvasó számítógép (lásd a 24. fejezetet!) meglehetősen drága kiegészítő berendezéseket igényel, és még így is csak korlátozott mértékben alkalmazható. Az írott, nyomtatott bizonylatok gépbe való beviteléhez pedig emberi munka szükséges, ami erősen idő- és költségigényes.

A számítógépes alkalmazások árába be kell még számolnunk a feldolgozási rendszer megtervezésének, megszervezésének, dokumentálásnak és bevezetésének költségeit is. Ez elsősorban szellemi munka. Ehhez hozzákapcsolódik a programok elkészítésének költsége, amelyben a szellemi munka mellett jelentős tényező a programok ellenőrzéséhez, kipróbálásához fölhasznált gépidő. Egy 4000 utasításból álló közepes méretű adott célra kidolgozott ügyviteli programrendszer elkészítése körülbelül egy emberévet vesz igénybe. (Ez nem azt jelenti, hogy egy ember egy évig dolgozik rajta, hanem azt, hogy egy 3–4 fős csoport 4–3 hónapot, összesen 12 hónapnyi munkát fektet bele elkészítésébe.) Ennek értéke (nem csupán a munkabér, hanem az erre rárakódó egyéb költségek) kb. 500 ezer Ft. Ehhez hozzáadódik még a program ellenőrzéséhez fölhasznált gépidő is, melynek nagysága 300 eFt körül van. Aki tehát egy ilyen rendszert akar saját igényeire szabva elkészíteni, annak ilyen mélyen kell a (vállalata, intézménye) zsebébe nyúlni. Ha a feladat típus jellegű, akkor lehet, hogy hasonló munkára már készítettek és forgalomba is hoztak olyan szoftverrendszert, amelyik ezt a problémát megoldja. Ennek ára általában tizede, vagy még kevesebb lehet, mint a megrendelésre készült termékeknek. Ezért a gondos gazdák igyekeznek minél több feladatot kész, vagy félkész szoftvertermékkel megoldani. Ez utóbbiakat tudatosan úgy tervezték, hogy a megoldásnak általában csak a vázát adják meg, a feladatra jellemző részleteket a felhasználó maga építheti be minimális munkával a programba.

Annak ellenére, hogy a szoftverfejlesztés módszerei egyre tökéletesednek, a számítógépes alkalmazások költségeinek jóval több mint a felét a szervezési-programozási munka teszi ki. Még egy új gép vételárának is legfeljebb a fele a hardver értéke. A többit a géppel együtt szállított, illetve az ahhoz megvett kész szoftverért fizetjük ki.

65.

ELŐTTÜNK AZ ÖZÖNVÍZ
(Elemi csapások, katasztrófák hatásának elhárítása)

Napjainkban a számítógép a vállalatok, intézmények adatainak legfőbb feldolgozójává és tárolójává vált. Ezen adatok és feldolgozások egy része a vállalat különböző erőforrásainak nyilvántartására, felhasználásának irányítására és ellenőrzésére szolgál (pl. pénzforgalom, raktárkészlet). Megsemmisülésük vagy meghamisításuk óriási károkat okozhat. Az adatok másik része a vállalat működéséhez nyújt segítséget. Ilyenek például a kereskedelmi vagy ipari titkokat, bizalmas személyi információkat tartalmazó adatok. A számítástechnikai biztonsági intézkedések egyik fő célja az, hogy a gép által szolgáltatott és tárolt információkat megvédjük az illetéktelen hozzáféréstől, módosítástól, megsemmisítéstől és nyilvánosságra hozatalától. A védelemnek ki kell terjednie a berendezések fizikai védelmén kívül az adatok, az alkalmazások és a vállalat érdekeinek a védelmére.

titkosság, biztonság

Ahogy nő az adatbankokban tárolt adatok mennyisége és fontossága, ugyanúgy nő az igény az iránt, hogy ezeknek az adatoknak titkosságát és biztonságát az adatbázis-kezelő rendszerek szavatolni tudják. A *titkosság* azt jelenti, hogy az adatbázisból bármely adathoz csak az arra illetékes személyek férhetnek hozzá. (Ezzel a kérdéssel az 57. és a 63. fejezetben foglalkoztunk.) A *biztonság* és az ezt megvalósító biztonsági eljárások, rendszabályok az adatoknak véletlenül vagy arra illetéktelenek által tudatosan történő módosítását, törlését vagy másfajta fizikai megsemmisítését küszöböli ki. Ebben a fejezetben az adatbiztonság témakörével foglalkozunk.

Az adatbiztonság megköveteli

- az adathordozók lopástól, tűztől, víztől és bármiféle egyéb természeti csapástól való védelmét;
- az adatok helyreállíthatóságát, mert minden elővigyázatossági rendszabály mellett is megsemmisülhetnek, meghibásodhatnak az adatok. Ennek érdekében gondoskodni kell a fontos adatok duplikálásáról és a változó adatok időnkénti kimentéséről és a változtatások feljegyzéséről.

Ezen feladatok megvalósításához szükséges biztonsági intézkedéseket két nagy csoportba osztjuk.

- A számítógépes rendszerekben magukban megvalósított biztonsági eljárások.
- A számítógépes rendszereken kívüli adminisztratív intézkedések.

A rendszeren belül megvalósított eljárások közé tartoznak azok a megoldások, melyeket a hardverben és/vagy az operációs rendszerben, illetve az egyes felhasználói programokban alkalmazunk a programok és az adatok védelme érdekében. Személyi számítógépeknél a rögzített lemezen levő adatok és programok védelménél ügyelnünk kell arra, hogy az operációs rendszerbe beépített ellenőrzéseket általában könnyen meg lehet kerülni a teljes operációs rendszer lecserélésével, ami igen egyszerűen megvalósítható. A szoftver módszerek a felhasználó személyét azonosítják, vagy az adatok formáját úgy módosítják (sifrírozzák), hogy azt csak a rejtjelezési módszer ismeretében lehet használható formába visszaállítani.

Az adminisztratív intézkedések közül a legfontosabb a személyzetnek és a számítógéphez való fizikai hozzáférésnek az ellenőrzése.

Az adatbiztonság jelentőségét mutatja, hogy az USA-ban 1985-ben 743 millió dollárt fordítottak erre a célra, mely 1990-re várhatóan megkétszereződik. Ennek az összegnek 95%-át számítógépes információs rendszerek biztonságának megőrzésére költötték.

A számítástechnikai rendszerek biztonságát minden fejlett országban külön törvény szabályozza. Magyarországon az erre vonatkozó 1/1981. (I. 17.) számú belügyminiszteri rendelet rendelkezik a számítástechnikai rendszerek titok-, vagyon- és tűzvédelméről. Hatálya kiterjed a számítástechnikai rendszert üzemeltető, feldolgozást megrendelő, valamint a feldolgozás eredményeit felhasználó szervekre, a műszaki, technológiai, szervezési-programozási eszközökre és a személyi jogok védelmére. Ennek megvalósítása érdekében minden számítástechnikát alkalmazó intézmény köteles kidolgozni a számítástechnikai szabályzatát.

Természetesen e biztonsági rendszert úgy kell megvalósítani, hogy az a legális felhasználók munkáját ne zavarja, az ő számukra a hozzáférést egyáltalában nem, vagy csak minimálisan nehezítse meg. Ezekkel a kérdésekkel a megfelelő helyeken már foglalkoztunk, itt most azt tárgyaljuk, mit tegyünk, ha a baj mégis bekövetkezett.

Különböző katasztrófák – tűz, víz, földrengés –, emberi hanyagság, rosszindulatú beavatkozás vagy szabotázs, különböző technikai problémák, mind okozói lehetnek egy számítógépes rendszer hosszabb-rövidebb ideig tartó kiesésének, ami gyakran együtt jár az adatok, programok, dokumentációk teljes vagy részleges megsemmisülésével. A számítógépes rendszer kiesése az üzemeltető intézmény számára gyakran azt jelenti, hogy kötelezettségeinek nem tud eleget tenni. Így komoly, esetleg végzetes erkölcsi és/vagy anyagi kára származik ebből. Ezért minden számítógépes rendszerre alapozó intézménynek fel kell készülnie arra, hogy a rendszer kiesése esetén is biztosítsa az információszolgáltatás folyamatosságát – esetleg elfogadhatóan csökkentett mértékben. Ennek érdekében előre meg kell tervezni, hogy a számítástechnikai szolgáltatások kiesésekor milyen helyettesítő megoldásokat lehet alkalmazni, illetve miként lehet a „katasztrófahelyzetről" a lehető leghatékonyabban és leggyorsabban a normális üzembe visszatérni.

A katasztrófa bekövetkezésének pillanatától a teljes helyreállításig terjedő szakaszt három részre oszthatjuk:

– válságszakasz,
– közbülső szakasz,
– helyreállítási szakasz.

A válságszakaszban mindennemű feldolgozás szünetel, nincs semmiféle lehetőség a rendszer üzemeltetésére. A fő cél ekkor a károk minimalizálása és a katasztrófa hatásának felmérése.

A közbülső szakaszban a *katasztrófa-helyreállítási terv* szerint a rendszertől függetlenül tárolt létfontosságú, kritikus adatokat és feldolgozásokat működésbe kell állítani akár az erre a célra előre kijelölt más számítógépen vagy számítóközpontban, vagy pedig ideiglenesen, manuális módon. Az előbbihez az kell, hogy a kritikus alkalmazáshoz szükséges adatok, programok, a futtatáshoz alkalmas teljes szoftverkörnyezettel együtt naprakész állapotban a számítóközponttól független helyen tárolva legyenek, és onnan a tartalék gépre azonnal át lehessen telepíteni. Ez a gyakorlatban csak akkor valósítható meg, ha a katasztrófa-helyreállítási tervben szereplő teendőket normális körülmények között mind végigpróbáljuk a saját rendszerszolgáltatásaink teljes kihagyásával.

katasztrófa-helyre-állítási terv

A helyreállítási szakasz folyamán fokozatosan visszaállunk az eredeti feldolgozási rendre a régi vagy esetleg új számítógépes környezetben.

A katasztrófa elhárítására szükséges felkészülés természetesen pénzbe és időbe kerül. Nem lehet minden adatot és alkalmazást, lehetőleg katasztrófabiztos módon duplikálva tárolni. Ezért lényeges ezekből a létfontosságúak kiválasztása, melyek nélkül az intézmény tökéletesen működésképtelenné válna. Ezeket mindenképpen így is tárolni kell. Azokat a fontos adatokat, melyeket a meglevőkből – bár esetleg csak elfogadható késedelemmel, nagy munkával, de elő tudunk állítani –, már nem szükséges ilyen nagy gondossággal őrizni. Ezek védelmére elegendő a szokásos másolatok készítése. Természetesen azt, hogy milyen információkat, feladatokat kell a létfontosságú kategóriába sorolni, az intézmény legfelsőbb vezetőségének kell megállapítania.

66.

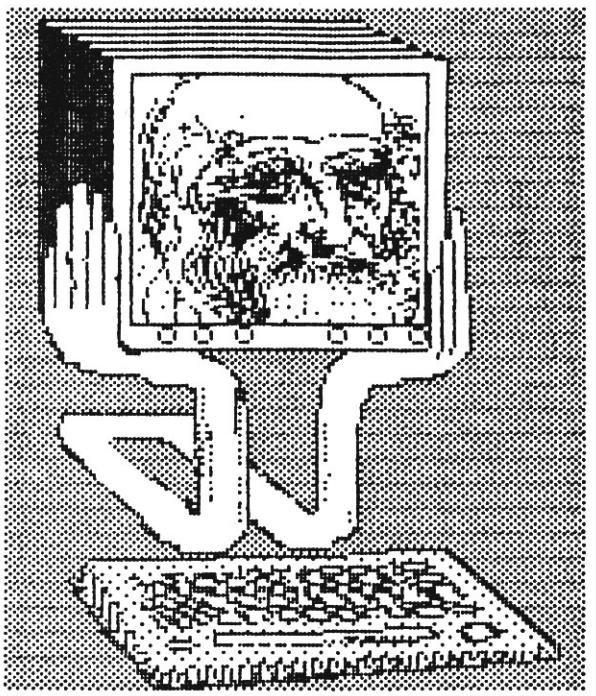
VERSENYBEN AZ EMBERI AGGYAL
(Szakértői rendszerek)

Az első számítógépek megépítése óta a tudósok és a mérnökök vágyálma egy olyan gép, amelyik nemcsak a betáplált programok utasításai szerint, hanem önállóan is tud gondolkodni, tanul a saját tapasztalataiból, és a mindennapi nyelvek kifejezéseivel teremt kapcsolatot az emberekkel. Turing angol matematikus módszert is javasolt, melynek segítségével eldönthető, hogy képes-e egy gép intelligens gondolkodásra. A mai napig még egyetlen számítógép sem teljesítette ennek a tesztnek a feltételeit. A japánok úgy tervezik, hogy a 90-es évekre kifejlesztett ötödik generációs számítógépek képesek lesznek már önálló, alkotó gondolkodásra is. De hogy ez valóban megvalósul-e akkorra, azt ma még nem tudhatjuk.

Olyan számítógép, amelyik általános értelemben vett hétköznapi módon tudna gondolkodni, és az ember előtt naponta felmerülő több ezer apróbb-nagyobb kérdésben „józan paraszti ésszel" döntene, azaz teljes mértékben leutánozná az emberi agyat és gondolkodásmódot, belátható időn belül valószínűleg nem fog készülni. Annak ellenére sem, hogy az előre programozott matematikai-logikai utasításokat a gép sok milliószor gyorsabban hajtja végre, mint az ember.

Ugyanekkor olyan gépek, olyan szakértői rendszerek, amelyek az élet egyes speciális területein gondolkodásban megközelítik az embert, már nagy számban léteznek. Jellemző példa erre a sakk, ahol versenyeket is rendeztek a számítógépek között. A játék összes szabályát egyszerűen, szabatosan meg lehet fogalmazni, a lépések egymásutánját, azok eredményét világosan meg lehet állapítani. Ennek ellenére a nemzetközi nagymesterekkel — legalábbis jelenleg — még a legnagyobb számítógépek sem vehetik fel a versenyt. De az átlagos sakkjátékost már a személyi számítógépek is le tudják győzni.

Azokat a számítógépes rendszereket, amelyek lehetővé teszik, hogy a gép részben, vagy egészen átvegye az embertől a döntést, az emberi gondolkodásmódot szimulálva ne csak mereven a belé programozott utasítások alapján dolgozzon (tehát intelligens segítőtársként viselkedjék), *szakértői rendszer*nek nevezzük. Minden ilyen rendszernek az alapja a *tudásbázis,* amely a témakörről eddig szerzett ismereteink összessége, a számítógép által értelmezhető formában tárolva. Ezek az ismeretek vonatkozhatnak a feladat természetére (leírják, mi a feladat) és arra is, hogy miképpen kell azt megoldani. Minden szakértői rendszer legfeljebb annyira lehet jó, mint az a tudásbázis, amelyre fölépül.



Mesterséges intelligencia

szakértői rendszer, tudásbázis

A szakértői rendszerektől elsősorban azt várjuk, hogy olyan problémákat tudjon megoldani, vagy megoldásukhoz hatékony segítséget nyújtani, melyeket különben szakértőknek kellene elvégezni. Ahogy azonban ismereteink bővülnek, úgy növekednek, változnak igényeink is. Ezért azt kívánjuk, hogy a szakértői rendszerek is önmaguktól be tudják építeni meglevő tudásbázisukba az új ismereteket, sőt segítsenek is bennünket a tudás átvételében. Azt is szeretnénk, hogy az ismeretanyagot a számunkra könnyen olvasható formában ábrázolják. Elvárjuk, hogyha cselekedeteink, döntéseink egy részét a szakértői rendszerek irányítják, akkor azok tanácsaikra, javaslataikra magyarázattal, indokolással szolgáljanak. Mivel ismereteink gyakran hiányosak (a rendelkezésre álló adatok nem elég pontosak), szeretnénk, ha a szakértői rendszerek pontatlan ismeretek alapján is döntenének vagy döntési változatokat javasolnának. Mivel a szakértői rendszerek alkalmazásánál a hangsúlyt nem a programozásra, hanem a tapasztalat, a tudás átadására óhajtjuk helyezni, azt kívánjuk, hogy a rendszerek értelmezni tudjanak a mindennapi élet nyelvén közölt mondatokat is. Napjaink szakértői rendszerei mindezen kívánalmakat még csak többé-kevésbé elégítik ki, de óhajaink közül már egyre több megvalósítható és meg is valósul.

Ahhoz, hogy mindezen követelményeknek megfeleljenek, a szakértői rendszereknek tartalmazni kell az alábbiakat:

- tények és ismeretek, gyakran szabályok formájában megfogalmazva,
- értelmező „gép" a tények és szabályok kapcsolatainak meghatározására,
- magyarázatgenerátor a döntések, döntési javaslatok indokolására,
- az ismereteket összegyűjtő „gép",
- természetes nyelvet értelmező processzor.

A szakértői rendszerek a probléma megoldásához kiindulhatnak abból, hogy a feladatokat általánosítják. Ezek megoldásait algoritmizálják, majd ezek közül a program próbálkozásokkal kiválasztja azokat, amelyek a konkrét feladat problémáját oldják meg. Az ilyen általános problémamegoldó rendszerek még a mai óriási sebességű számítógépeken is lassan dolgoznak, mert rendszerint nincs lehetőségünk arra, hogy felhasználjuk a feladat megoldására rendelkezésre álló speciális eljárási ismereteinket. (A sakkozó gép minden lépésnél nagyságrendekkel több lehetőséget vizsgál meg, mint egy nagymester, aki az elvileg lehetséges változatok többségét eleve elveti, mint érdektelent.) A speciális feladat megoldására vonatkozó ismereteink hiányában előfordulhat, hogy először rossz úton indulunk el, és a problémamegoldónk csak sok téves próbálkozás után jut el a helyes eredményhez.

A másik kiindulási lehetőség az, hogy az általánosság keresése helyett összegyűjtjük az adott speciális feladat megoldására rendelkezésre álló teljes ismeretanyagot. Ebből keressük ki azt, vagy azokat a megoldásokat, amelyek a konkrét feladatra illenek. Lényegében ez a módszer vezetett el az első, gyakorlatban is működő szakértői rendszerekhez. Az ismeretanyag, a tudásbázis összegyűjtése igen nagy szellemi befektetést igényel. Létrehozásához szükség van egy „tudásmérnök"-re, aki elemzi a szóban forgó feladatot, specifikálja a megfelelő „értelmezőgépeket", majd együttműködve a feladat szakértőivel és programozóival, felépíti a rendszert. Ez a munka rendszerint több évet is igénybe vesz.

A fenti két lehetőség szintézise valósul meg napjaink szakértői rendszereiben. Az alapgondolat az, hogy a különböző feladatoknak vannak közös követelményei, melyeket egy „szakértőirendszer héj"-ban meg tudunk fogalmazni. Ehhez a célhoz adjuk aztán hozzá az adott feladatra vonatkozó speciális ismereteinket, hogy az elvileg lehetséges megoldások közül a reménnyel kecsegtetőket vizsgálja meg először a rendszer, így gyorsabban jusson el a megoldáshoz.

Mivel a szakértői rendszerek logikája különbözik a hagyományos programozás logikájától (utóbbiban mindig egyértelműen meg kell mondani, hogy milyen feltételek teljesülésekor milyen utasításokat kell végrehajtani), a szakértői rendszerek programjainak elkészítéséhez külön programozási nyelveket dolgoznak ki. Ezek közül a legismertebb a PROLOG és a LISP. A PROLOG-nak a kifejlesztésében és alkalmazásában jelentősen közreműködtek magyar szakemberek is.

A szakértői rendszerek már több területen igazolták hatékonyságukat. Egyik igen fontos alkalmazási lehetőségük: orvosi diagnózisok felállítása, amiről a 60. fejezetben is írtunk.

Miután a gépet megtanították a különböző betegségek tüneteire, azokra a kérdésekre, melyeket ezekhez kapcsolódva a betegeknek fel kell tennie és azokra a vizsgálatokra, melyeket el kell végeztetni, hogy a kapott válaszok és eredmények alapján pontosabb, végül pedig egyértelmű legyen a diagnózis és a javasolt gyógymód, a gépdoktor fölveszi a versenyt a legjobb diagnosztával is. Természetesen azt a kapcsolatot, amikor a vizsgálat végén a doktor bátorítóan megveregeti a beteg kezét és azt mondja: „Nyugodjék meg, Mari néni, nincsen semmi baj!", egy számítógép sem fogja tudni megteremteni.

Még akkor sem, ha annyira intelligens lesz, hogy azt is tudja majd, mekkora hálapénzre számíthat ebből a kézből – az üzemeltetője.

67.

ÁLDÁS ÉS ÁTOK
(A növekedés határai)

Minden tudományterületen előbb-utóbb felmerül a kérdés, hol vannak az adott terület növekedésének határai. „A határ a csillagos ég!" – tartja a mondás, de talán a csillagászatot leszámítva, minden terület ennél alább kell, hogy adja.

A látványosan fejlődő tudományterületeken ez a kérdés nemcsak a szakembereket, hanem a közvéleményt is foglalkoztatja. Így van ez a számítástechnika esetében is, hiszen egyre fantasztikusabbnak tűnő méretcsökkenésekről és teljesítménynövekedésekről értesülhetünk nemcsak a szaklapokból, hanem a hírközlés mindennapos csatornáiból is.

A számítógépek képességeivel kapcsolatos kérdések először – érdekes módon – még az első számítógép megjelenése előtt megfogalmazódtak. 1936-ban (vagyis egy évtizeddel az első mai értelemben elektronikusnak tekinthető számítógép megépítése előtt) Turing angol matematikus mondta ki, hogy automatikusan működő számoló automatával elvben minden feladat megoldható, amelyre úgynevezett effektív eljárást tudunk adni. Az effektív eljárást mai szóhasználattal algoritmusnak nevezzük, vagyis az elvi megoldhatóság feltétele az, hogy a probléma véges számú részlépés sorozatával leírható legyen.

1958-ban Novikov szovjet matematikus talált olyan számelméleti problémát, amelyről bebizonyította, hogy nem algoritmizálható. A ma szakemberét azonban elsősorban már nem ez érdekli, hanem az, hogy milyen áron oldható meg egy-egy probléma. A számítógépek alkalmazhatóságát ugyanis alapvetően két tényező befolyásolja:

– mekkora kapacitás és milyen sebesség szükséges a probléma megoldásához,
– mekkora költséggel jár a probléma megoldása.

Mindkét tényező végső fokon költségeket takar, de ezeknek a kérdéseknek csak akkor van értelmük, ha felteszünk és megválaszolunk egy harmadikat is:

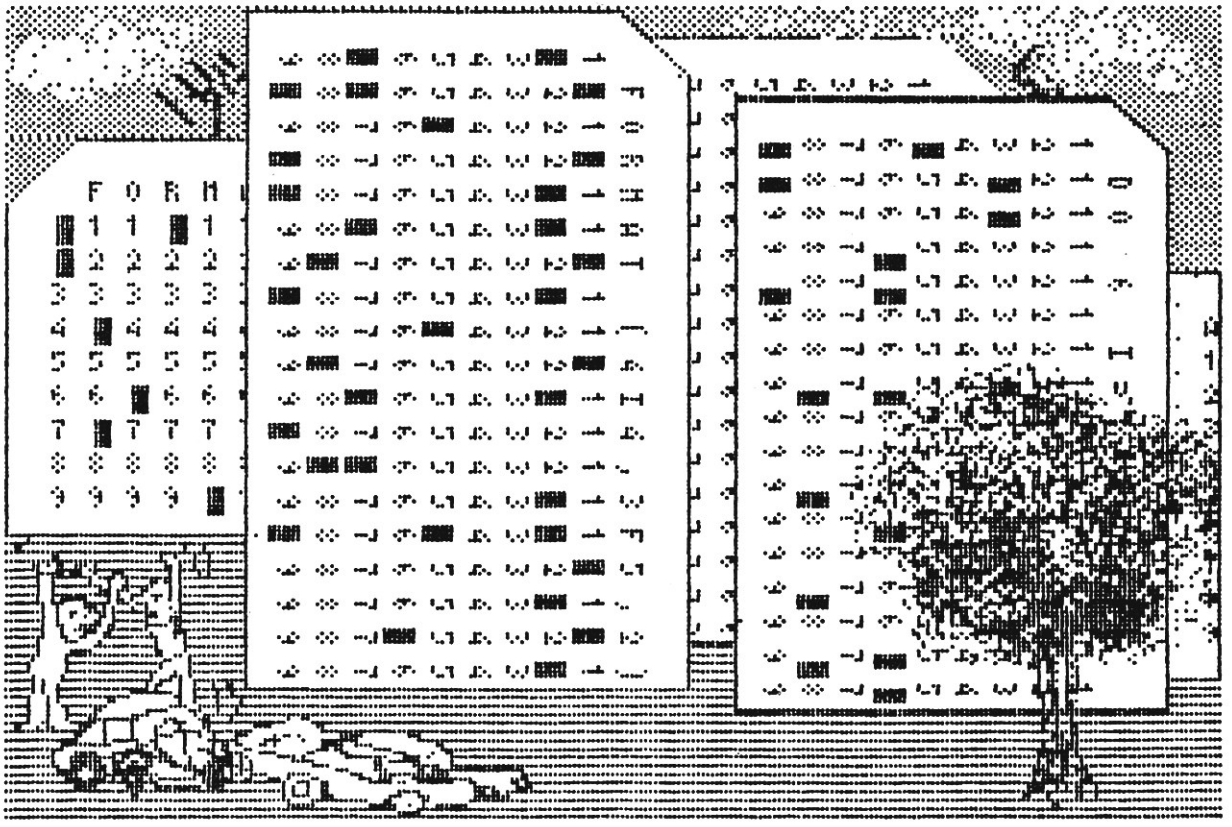
– milyen mértékben fontos számunkra, hogy a probléma megoldott legyen.

Ha ugyanis a feladatot mindenáron meg kell oldani (gondoljunk csak az űrkutatással kapcsolatos, csak számítógéppel megoldható problémákra), a „mibe kerül" egészen más színezetet kap, mint például egy statisztikai adatgyűjtő rendszernél, ahol értelmes kompromisszumok köthetők a költségek csökkentése érdekében.

Magyarországon minden állampolgár rendelkezik személyi számmal. Tudjuk, hogy ennek segítségével bárki azonosítható. De talán kevesen tudják, hogy az ország összes lakosának személyi számát, nevét és lakcímét (nem többet!) magában foglaló nyilvántartás olyan méretű, hogy *egyszerre* a hazánkban levő legnagyobb számítógéppel sem kezelhető. S ha valakinek olyan ötlete támadna, hogy ezt a nyilvántartást néhány további adattal kellene kiegészíteni, a világ jelenleg legnagyobb számítógépe sem volna képes arra, hogy a teljes állományt *egyidejűleg* kezelni tudja.

S bár egyre nagyobb teljesítményű (tárkapacitású, műveletvégző sebességű) gépek jelennek meg a piacon, a távlatokat vizsgálva elsődlegesen mégsem ez érdekes, hanem az, hogy mi mindenre lesz képes a jövő számítógépe.

A mesterséges intelligenciával kapcsolatos kutatások (lásd a 66. fejezetet) jelentik azt az irányt, ahonnan az elkövetkező években (évtizedekben) a legfontosabb eredmények várhatók. És ugyanitt fognak felmerülni a korlátok is. Nem egetverő problémák ezek, hanem aprók, amelyek azonban figyelmeztető erejűek. Az irodában takarító robot, amelyet számítógép vezérel, a papírkosár teljes tartalmát akkor is a kukába fogja üríteni, ha abban egy nagy értékű XX. századbeli (még mutatókkal működő) zsebóra ketyeg, mert az egyik tisztviselő beleejtette.



Természetes(?) intelligencia

Sokat cikkeznek manapság arról, hogy a tervezőmunkában milyen sok segítséget nyújtanak a számítógépek, mennyi felesleges „rabszolgamunkától" kímélik meg a tervezőmérnököket. Nos, az első próbálkozások egy angol repülőgép-tervező irodában nem jártak osztatlan sikerrel. A tervezőmérnökök ugyanis ahhoz voltak szokva, hogy hosszas gondolkodás, vázlatkészítés, újragondolás után, két-három hónapos munkával megoldják a kapott feladatot. A számítógép az egyes közbülső változatokról pár másodperc alatt eldöntötte, hogy miért hibás. A két-három hónapos fázishoz szokott tervezők képtelenek voltak néhány óránként újabb és újabb változatokat előállítani. Képtelenek voltak elviselni a „rabszolgamunka" hiányát, amely úgy látszik az alkotó munka szerves része volt számukra.

A televízióból közismert, hogy a meteorológiai előrejelzésekben már ma is fontos szerepet töltenek be a számítógépek. Nos, ha jelenleg egy 24 órás előrejelzés meghatározása egy számítógéppel mondjuk egy óra alatt elvégezhető, nyilván nem jelent fejlődést, ha ez az idő fél órára csökken. Az igazi eredmény az lenne, ha sikerülne egy hétre előre jelezni az időjárásváltozásokat azzal a valószínűséggel, amellyel ma a napi előrejelzések készülnek. De ehhez nem csupán a mainál jobb számítógépek kellenek.

A felsorolt példák rávilágítanak arra a nagyon fontos tényre, amelyet nem mindenki lát ma tisztán: az alkalmazások korlátai még nagyon hosszú ideig nem a mindenkor ren-

delkezésre álló technikában, hanem bennünk, emberekben lesznek. Ma még nem látjuk tisztán, hogy az újabb technika milyen újabb, eddig ismeretlen korlátokat hoz a felszínre az egyénből és a társadalomból.

Ugyancsak sok szó esik manapság az elektronizált társadalomról. A tudósok egyetértenek abban, hogy a számítástechnika valamennyi áldása csak a társadalom teljes elektronizálásával valósulhat meg. Abban azonban már nem ennyire egységes a megítélés, hogy mindez milyen hatással lesz az emberi életre. Humánusabb lesz-e egy racionálisabb társadalom? Az emberi boldogulást biztosítja-e a csökkenő munkaidő? Megtalálja-e mindenki a helyét egy ilyen társadalomban?

Nehéz kérdések ezek, amelyeket csak az idő fog megválaszolni. De úgy gondoljuk, és bízunk abban, hogy rajtunk is múlik ezeknek a válaszoknak a milyensége.

68.

NAGY NEVEK – KIS TÖRTÉNELEM

A diszkóműsorokhoz hozzátartozik egy kis nosztalgia is: „gyerekek, ismerjétek meg régi idők nagy muzsikusait!" Az egykori BEATLES-en nevelkedett nemzedék nevében Tömpe Zoltán ajánlja most a BIT-LES olvasóinak, ismerkedjenek meg a számítástechnika történetének legnagyobbjaival.

Blaise PASCAL (1623–1662)

Francia matematikus, fizikus, filozófus. A projektív geometria egyik alaptételét Pascal-tételnek nevezzük. Az ő nevét viseli a nyomás mértékegysége. Ő értelmezte és alkalmazta először a közlekedőedények törvényét. Fizikai kísérletei során megállapította a légnyomásnak a magasságtól való függését. A folyadékok és a levegő fizikájával kapcsolatos munkássága nagy lépésekkel vitte előre a kísérleti kutatásokat. Pascal volt a teljes indukció módszerének első következetes alkalmazója. Vele indult meg a valószínűségszámítás. Pascal a megalkotója az első modern értelemben vett mechanikus számológépnek.

1640-ben látott hozzá egy számológép kialakításához, azért, hogy segítse apja adószedő munkáját Rouenban. Az első működő modell 1642-ben készült el, majd a következő tíz évben további 50 darabot készített. A gép tulajdonképpen nyolc tárcsával felszerelt kis doboz volt. (A tárcsák hasonlóak a mai telefon számtárcsához.) Mindegyik tárcsa egy dobhoz kapcsolódott, amely egy regiszterablakon keresztül megjelenítette a számjegyeket. Pascal önálló fejlesztésének eredménye volt egy fogazott lánc a forgódobok között, amely átvitte a forgómozgást az egyik dobról a következő, nagyobb helyi értékűre, amikor a számítás eredménye ezt igényelte. A gép közvetlenül végezte el az összeadást és kivonást. A szorzás és osztás többszörös összeadással és kivonással volt végrehajtható.

Gottfried Wilhelm von LEIBNIZ (1646–1716)

Német filozófus és matematikus. Apja könyvtárában ragyogó tudásra tett szert, és mielőtt 15 éves lett volna, már a lipcsei egyetemre került. 17 éves korában diplomázott, és húszévesen jogi doktor lett. Kezdetben jogászi és diplomata pályán működött. 1672-ben mint diplomata került XIV. Lajos Párizsába, ahol matematikai pályafutása legtermékenyebb négy évét töltötte. Descartes és Pascal nyomán tanulmányozta a matematikát. Pascal számológépe élénk érdeklődést váltott ki Leibnizből. Módosította, továbbfejlesztette Pascal gépét, amelyet kerekeken mozgatható kocsival és késleltetett mozgató mechanizmussal látott el, így a gép közvetlenül alkalmas lett szorzásra és osztásra. 1673-ban Leibniz fontos felfedezéseket tett a differenciálszámítással kapcsolatban. 1675-ben felfedezte, hogy az integrálás összeadási folyamata tulajdonképpen a differenciálási művelet ellentéte. Newton ezt már 1660-ban, tehát 15 évvel korábban felismerte, de erről

Leibniz nyilvánvalóan nem tudott, illetve nem tudhatott. 1676-ban elhagyta Párizst, és Hannoverbe költözött, ahol hátralevő 40 évében történelemmel, filozófiával, teológiával, diplomáciával és tudományos levelezéssel foglalkozott. Közben megszakításokkal számológépén is dolgozott. Már 1668-ban tervbe vette, majd 1700-ban megalapította a berlini Tudományos Akadémiát. Leibniz a mai műveleti jelek meghonosítója (+, −, ., :). Továbbá az ő nevéhez fűződik a differenciálszámítás szimbolikájának megalkotása. Ő tekinthető a szimbolikus logika előfutárának.

Charles BABBAGE (1791–1871)

Babbage a modern intellektuális történelem egyik legkülönösebb alakja. Egyik történetírója jellemző módon „a lobbanékony géniusz"-nak nevezte. Felső középosztálybeli angol családban született és ez óriási intellektuális és társadalmi előnyt jelentett számára. Babbage kora szellemi életének középpontjába került. Már egyetemi évei alatt közeli kapcsolatba került a nagy csillagásszal, John Herschellel és Georges Peacock-kal. 1810-ben szerzett diplomát a cambridgei egyetemen, amely akkor erősen matematikai irányultságú volt. Egy anekdota szerint 1822-ben Herschell és Babbage csillagászati számításokat ellenőriztek, amikor Babbage mérgében állítólag azt mondta: „Adná Isten, hogy ezeket a számításokat gőzgéppel lehessen elvégezni". Nagyon érdekes, hogy Babbage ösztönösen az akkori kor csúcstechnológiáját képviselő gőzgépekre gondolt. Még ugyanebben az évben, tehát 1822-ben Babbage levelet írt a Royal Society akkori elnökének, a nyomtatott matematikai táblázatok kiszámításának fárasztó monotonitásáról, és elküldött egy tudományos értekezést az ilyen számítási munkák automatizálásáról. A Royal Society kiküldött egy bizottságot, amely végül is javaslatot tett, hogy Babbage számológépe építéséhez kapja meg a szükséges állami támogatást. A kormányzat úgy gondolta, hogy ez a számológép a tengerészet számára is igen értékes lehet a különböző hajózási táblázatok készítésénél.

A terv ily módon sínre került, és 1823-ban megkapta a szükséges állami támogatást. Babbage nagy ambícióval látott hozzá differencia gépe építéséhez. A differencia gép tulajdonképpen a számítási műveleteket próbálta mechanizálni és a nyomtatott matematikai táblázatok készítését segítette volna elő. Végül is tízévi munka után kudarcot vallott. 1833-ban fölhagyott a differencia gép építésével. Ez nagy személyes csalódás volt Babbage életében, pedig a differencia gép brilliáns kísérlet volt. 1827-ben nevezték ki professzornak Cambridge-ban Newton egykori tanszékére. 1833-ban, amikor a differencia gép fejlesztésével felhagyott, Babbage kigondolta főművét, az analitikus gépet. Ez elvileg már általános célú számítógép volt. Ez a gép életcéljává vált Babbagenek, és egészen 1871-ben bekövetkezett haláláig dolgozott rajta. E gép alapötlete merőben különbözött a korábbi differencia géptől.

Ekkor már meglehetősen világosan látott több olyan elvet, amely a modern számítógépeket jellemzik. Az analitikus gép decimális volt, a számokat fogaskerekek tárolták, tíz különböző helyi értéken elhelyezve. Egy központi „malom" forgatta volna az egész mechanizmust. A malomba mindig azokat a mennyiségeket kellett volna bevinni, amelyekkel éppen valamilyen műveletet kellett végezni. Az elképzelés szerint az analitikus gép tárolási kapacitása ezer darab, 50 decimális jegyből álló szám volt. Az analitikus gép két lyukkártya-csomagot használt, a műveleti kártyákat és a változó kártyákat. Ebben az elrendezésben, ha bármilyen formulát ki akarunk számítani, a műveleti kártya csomagját úgy kell egymás után elhelyezni, hogy azok a műveleteket olyan sorrendben tartalmazzák, ahogy a formulában előfordulnak. Ezután egy másik kártyacsomagot kell összeraknunk,

amely a változókat behívja a malomba, abban a sorrendben, ahogy dolgozni kívánunk velük. Babbage, korát messze megelőzve feltalálta a számítógépet, pedig kora akkor még tulajdonképpen nem igényelte igazán a számítógép felfedezését. Sőt, a technológia akkori fejlettségi szintje nem igazán támogatta Babbage kutatásait.

A modern elektronika felhasználására Babbage akkor még nem gondolhatott. Babbage érdeklődési köre rendkívül széles volt. Kezdve az olyan eszközöktől, mint a vasúti kocsikra szerelhető dinamométer, a búvárharang és a szemtükör, egészen a rendszerelemzésig terjedt. Foglalkozott matematikával, közgazdaságtannal és politikával is. De fő érdeklődése a számítógép volt. Babbage Anglia és a kontinens legjobb társadalmi- és szellemi köreiben mozgott, barátságban volt Darwinnal, Dickenssel, Fourier-val, Humboldttal, Millel és Thackeray-vel, hogy csak néhány nevet említsünk. Nagyon jellemző volt Moulton visszaemlékezése: „Keresztülvezetett dolgozószobáin. Az első szobában a differencia gép darabjait láttam. Érdeklődtem, hogy áll vele e pillanatban. Nem fejeztem be – válaszolta – mert, miközben dolgoztam rajta, rájöttem az analitikus gép elvére, amely mindent tud, amire ez képes, sőt még sokkal többet is. Az ötlet annyival egyszerűbb volt, hogy több munkát jelentett volna befejezni a differencia gépet, mint teljes egészében megépíteni ezt a másikat, az analitikus gépet. Megkérdeztem: láthatnám-e magát az analitikus gépet. Nem fejeztem be, mert rájöttem, hogyan lehetne ugyanazt másképpen, sokkal hatékonyabban csinálni, és így értelmetlenné vált, hogy a régi módon folytassam. Ezután egy harmadik szobába mentünk, és itt egy szerkezet szétszórt darabjai hevertek. Kérdésemre Babbage azt felelte: Még nem csináltam meg, de dolgozom rajta. Amikor néhány évvel később meghalt, nemcsak hogy semmilyen megépített gép nem maradt utána, hanem egy bizottság azt volt kénytelen megállapítani, hogy minden túlságosan kezdetleges állapotban volt, semhogy valamilyen hasznos célra föl lehetett volna használni."

George BOOLE (1815–1864)

Boole Angliában született, alsó középosztálybeli családban, olyan társadalmi rétegben, amelynek gyermekei nem nagyon jutottak el az egyetemre. Boole az elemi iskola elvégzése után teljesen önképző alapon tanult tovább. Megtanult latinul és görögül, majd megismerte a matematika alapjait. 16 éves korában munkába kellett állnia, hogy eltarthassa szüleit. Két iskolában is segédtanári állást vállalt, és tanított négy éven keresztül. Ez alatt megtanult franciául, németül és olaszul. Sikerült elolvasnia és megértenie korának valamennyi nagy matematikusát. Húszéves korában Boole saját iskolát nyitott, és maga tanított felső matematikát. 1849-ben, annak ellenére, hogy nem volt formális képzettsége, kinevezték matematika professzornak az újonnan alakult Queens College-be, Corkban, Írországban. Mintegy ötven tudományos értekezést írt, és négy könyvet, melyből kettő a nevével fémjelzett matematikai logikáról szólt. Felesége, Mary Everest, Sir George Everest, a Mount Everest névadójának unokája volt.

Bár Boole számos lényeges dolgot alkotott a matematika területén, halhatatlanságát mégis azzal a két könyvvel szerezte, amely később Boole-algebrának elnevezett matematikai logikával foglalkozott. 1848-ban jelent meg „A logikai matematikai elemzése" című könyve, majd 1854-ben a nagy műve, melynek címe „A gondolkodás törvényeinek kutatása, amelyen a logika és a valószínűség matematikai elmélete alapul". Ezekkel a műveivel Boole a modern szimbolikus logika megalapítójává vált. A logikát egyszerű számítások sorozatára redukálta. Határozott erőfeszítéseket tett a logika és a matematika egyesítésére, és hatására egész szimbolikus logikai iskola alakult. Életében is sikeres volt, de a Boole-algebra lángja csak fél évszázaddal halála után, 1910–13 tájékán gyulladt meg. Vagyis

fél évszázadig Boole gigantikus műve egyszerű érdekesség maradt. Boole felfedezése, nevezetesen, hogy az algebra szimbolizmusa használható a logikában is, jelentős lökést adott a XX. század tudományának.

Ma a Boole-algebra nemcsak a matematikai logika, hanem a valószínűség-elmélet, a háló-elmélet, a síkgeometria, és az információelmélet számára is fontos. Boole-nak a logikában kifejtett tevékenysége tette lehetővé a logika későbbi kutatóinak, köztük Neumann Jánosnak munkásságát is. Neumann műve pedig nélkülözhetetlen volt a modern számítógépek kifejlesztésénél. Minthogy Boole megmutatta, hogy a logika igen egyszerű algebrai rendszerekre redukálható, így lehetségessé vált, hogy olyan szerkezeteket tervezzenek, melyek végre tudják hajtani a szükséges logikai feladatokat.

Hermann HOLLERITH (1860–1929)

Hermann Hollerith a lyukkártyás adatfeldolgozás feltalálója. Hollerith szülei a történelmi jelentőségű 1848-as esztendő politikai forrongásai következtében hagyták el Németországot. Hollerith Buffaloban született, és a Columbia egyetem bányászati szakát végezte el. Doktori értekezésében egy saját maga szerkesztette táblázatkészítő rendszert tárgyalt. 1879-ben szerzett diplomát a Columbia egyetemen, majd ezután a Népszámlálási Hivatalnál helyezkedett el. A Népszámlálási Hivatalt az 1880-as népszámlálás lebonyolítására hozták létre. Hollerith itt kezdett gondolkodni egy olyan gépen, amely megkönnyíti az adatok mechanikus feldolgozását. Hollerith 1879-től 1883-ig tartozott a Népszámlálási Hivatal állományába, majd később a szabadalmi hivatalban kapott állást. Ettől kezdődően élete végéig a népességi statisztikák táblázatba foglalására alkalmas gépének kifejlesztésén dolgozott. Erre 1889-ben meg is kapta a szabadalmi oltalmat.

E munkája mellett kiépített egy kereskedelmi szervezetet, a Tabulating Machine Companyt. 1911-ben ez a társaság átalakult Computer Tabulating Recording Companyvé, amelyhez 1914-ben lépett be Thomas J. Wattson. Ebből a társaságból lett később, 1924-ben, Wattson mesteri irányításával az IBM (International Business Machine Corporation). 1890-től egészen a világháború kitöréséig, vagyis negyed évszázadig Hollerith és cége monopolhelyzetben volt a lyukkártyás adatfeldolgozásban. Hollerith összesen mintegy ötvenféle különböző szabadalommal rendelkezik, különböző lyukkártyás módszerekre és berendezésekre. Számos alkalmazási területen vezette be a lyukkártyás adatfeldolgozást. Ilyenek például a népszámlálás, egészségügyi statisztika, vasúti adatok feldolgozása, közérdekű elszámolások feldolgozása, raktárkészlet-elszámolás, gyári költség elszámolása stb. Sok elképzelése máig is érvényes. A lyukkártyás kód (Hollerith kód) 1901 óta használatos.

Thomas J. S. R. WATTSON (1874–1956)

A középiskola elvégzése után, 1892-ben munkába állt, mint könyvelő, heti 6 dolláros fizetésért. Nem sokkal később varrógépekkel és hangszerekkel kereskedett, majd belépett az NCR (National Cash Register) céghez, mint üzletkötő. Négy év múlva az NCR kinevezte őt a cég Rochester városi vezetőjének. Újabb négy és fél év múlva már ő az NCR általános kereskedelmi vezetője. Wattson 1913-ban otthagyta az NCR-t, és nősülés révén a Computing Tabulating Recording Company elnöki székében találta magát. 1913-tól haláláig, azaz 43 évig töltötte be ezt a pozíciót. Közben 1924-ben a cég új nevet kapott,

és IBM (International Business Machine) néven lett a világ legnagyobb számítástechnikai vállalata. Wattson vezetése alatt a cég a világ vezető vállalata lett a lyukkártyás adatfeldolgozás területén. 1934-ben az IBM vezette be először a piacra az elektromos írógépet, amivel óriási sikert aratott. Szintén Wattsonnak köszönhető, hogy a cég 1948-ban elindult a számítástechnika irányába. Kezdetben elektronikus programozható számológépeket, majd az 50-es évektől elektronikus adatfeldolgozó számítógépeket gyártottak, forgalmaztak.

A világ teljes számítógépforgalmának 35—40 százalékát az IBM egymaga teszi ki, de volt már 50 százalék fölötti is a részesedése. A cég abszolút vezető szerepét mutatja, hogy csak az a számítástechnikai újdonság tud jelentős piaci sikert aratni, amit az IBM is átvesz. (Pl. miniszámítógép, PPC — akkor terjedtek el, amikor már az IBM is gyártotta.

Howard AIKEN (1900—1973)

Indianapolisban nőtt fel, majd 1923-ban elvégezte a Wisconsin egyetemet. Ezután munkába állt, a Madison Gas Companynál helyezkedett el. 1935-ben ismét az iskolapadba ült, és 1939-ben doktorált a Harvard egyetemen. Doktori disszertációjához sok bonyolult, nem lineáris differenciál-egyenletet kellett megoldjon, és ez vezette el ahhoz, hogy megvizsgálja ezen számítások gépi elvégzésének lehetőségét. 1937-ben tanulmányt írt „Automatikus számológép" címen. Akkoriban a Harvard egyetemen nem volt a legingerdúsabb környezet az ilyen jellegű kutatások számára. Ezért azután tanárai beajánlották Aikent Sir Thomas Wattsonhoz, az IBM-nél, ahol adottak voltak a technológiai feltételek az ilyen jellegű kutatásokhoz. 1939-ben az Egyesült Államok haditengerészete megbízást adott az IBM-nek egy automatikus számítógép megépítésére. Ezt a munkát Aiken irányította. 1944-ben elkészült a Harvardon a Mark I. jelű számológép. Ezután még több számológépet fejlesztett Aiken az Egyesült Államok haditengerészetének és légierejének megbízásából. A Mark II. 1946-ban, a Mark III. 1950-re, a Mark IV. pedig 1952-re készült el. A Mark I—III. decimális rendszerben működő elektromechanikus eszközök voltak, meglehetősen lassú műveletvégzési sebességgel.

Aiken elsődleges jelentősége abban van, hogy ő alapozta meg a Harvard Computation Laboratoryt. Ezzel rendkívül ütőképes környezetet hozott létre a Harvard egyetemen, amely egy sor tehetséges fiatal szakember elsőrendű iskolájává vált. Aiken 1961-ben vonult vissza a Harvard egyetemről és Floridában saját vállalatot hozott létre Aiken Industries néven. Közben a Miami egyetem meghívott professzora lett, és aktív segítséget nyújtott ahhoz, hogy a miami egyetemen is beindult egy hatékony számítástechnikai tudományos program,és felépült a számítóközpont.

Norbert WIENER (1894—1964)

Az egyik legjelentősebb amerikai matematikus. 1948 óta neve összefonódott a kibernetikával. Ezt a nevet adta ugyanis a tudományos szenzációt kiváltó könyve címének, melyben a kibernetikát úgy határozza meg, hogy „az élő szervezetekben és a gépben történő kommunikációnak és vezérlésnek az elmélete". Tarján Rezső szerint a kibernetika egyik középponti fogalma a visszacsatolás, valamint a megbízhatóság. A kibernetika egy sor új kérdést vizsgált, és egy sor régi problémát új módon vetett fel.

A kibernatika olyan új, komplex tudományos irányzat, amely bonyolult, magasan szervezett rendszerek felépítését és viselkedését tanulmányozza a természettudományok szigorú módszereivel, elsősorban elméleti, de ahol lehet kísérleti módszerekkel is. A kibernetika a tudományok történetében az első tudatos kísérlet azoknak az organizációs elveknek a műszaki konstrukciós értelemben vett gyakorlati alkalmazására, melyek a legmagasabb szervezett rendszerekben, az élő szervezetekben megfigyelhetők.

Wiener megalapozta a filozófiai kapcsolatot a mechanika és a matematika között. Munkája nem kapcsolódott valamely konkrét számítógép kifejlesztéséhez. De elméletileg elősegítette az automata kutatásokat. Wiener sokat foglalkozott a számítógép, az ember és a társadalom összefüggésével.

Wallace J. ECKERT (1902–1971)

Wallace John Eckert a modern asztronómia talán legnagyobb alakja volt. Az asztronómia numerikus csillagászatot jelent, és nem tévesztendő össze az asztrológiával. Eckert nevéhez fűződik a gépi számítás bevezetése az asztronómiában. 1926-ban szerzett diplomát az Oberlin College-ben. 1931-ben a Yale egyetemen az asztronómiai tudományok doktora lett. Azonban már ezt megelőzően is 1926-tól a Columbia egyetem statisztikai célokra IBM lyukkártyás gépeket vásárolt. Eckert ekkor találta szembe magát azzal az elvi lehetőséggel, hogy gépeket lehet használni különböző számítások megkönnyítésére. Thomas Wattson, az IBM akkori vezérigazgatója, 1929-ben megalapította a Columbia egyetem statisztikai hivatalát. Itt Eckert 1929 és 1933 között lyukkártyás gépeket használt asztronómiai adatok interpolálására, a megfigyelt adatok egyszerűbb feldolgozására.

Eckertnek döntő hatása volt abban, hogy az IBM a lyukkártyás adatfeldolgozás felől megindult az elektronikus számítástechnika felé. 1945-ben kinevezték az IBM elméleti tudományos főosztályára főosztályvezetőnek, és ezzel párhuzamosan ő lett a Wattson Scientific Computing Laboratory igazgatója is. Az 1960-as években Eckert részt vett a Nasa Surveyor, Lunar Orbiter és Apolló programjaiban. 1967-ben nyugalomba vonult az IBM-től, majd 1970-ben a Columbia egyetem professzori katedrájától is.

NEUMANN János (1903–1957)

A XX. századi matematika legendás figurája. Számtalan anekdota kering gyors gondolkodásáról, abszolút emlékezőképességéről, matematikai zsenijéről, nyelvérzékéről, humorérzékéről. Nagyon jelentős tudományos eredményeket ért el a logikában, a kvantumfizikában, a nagy sebességű számológépek elméletében, a meteorológiai számításokaban, a matematikai játékelméletben és stratégiában, sőt a közgazdaságtanban is. Munkássága az összes általa művelt területen, önállóan is óriási tudományos teljesítmény, ezért Neumann János különleges szerepet tölt be a modern tudományban. 1911 és 1921 között a fasori lutheránus gimnáziumba járt Budapesten. Már 18 éves kora előtt megírta első matematikai tanulmányát. Apja ellenezte, hogy matematikus legyen, ezért 1921 és 1923 között Berlinben, majd 1923 és 1925 között Zürichben vegyészetet tanult és vegyészmérnöki diplomát szerzett. Hazaérkezése után egy éven belül, 1926-ban Budapesten matematikai doktorátust szerzett. A húszas évek végén nemzetközi hírnevet szerzett magának matematikai elméleteivel. 1930-ban meghívták vendégtanárnak az Egyesült Államokba, a Princeton egyetemre. 1940-ig elsősorban elméleti matematikusként vált híressé, 1940 után pedig az

alkalmazott matematikában alkotott maradandó eredményeket. 1944-ben az atombombával kapcsolatos számítási problémák vezették el Howard Aiken gépéhez, a Mark I-hez a Harvard egyetemre. 1944 nyarának végén Hermann Goldstine közölte Neumannal, hogy a világ első elektronikus számítógépe Mauchly és Eckert irányítása alatt hamarosan elkészül a Moore Schoolban. Neumann János 1944 augusztusában látogatta meg először az ENIAC programot.

Az ENIAC akkor már többé-kevésbé kész gép volt, így Neumann csak a gép továbbfejlesztésére tett 1947-ben Adele Goldstine-nel közös javaslatot. Az áttervezett gép könnyebben volt programozható. Neumann fogalmazta meg az EDVAC előzetes tervét (1944-ben), amelyben a mai modern számítógépek szinte valamennyi lényeges funkcióját leírja. Miután 1945-ben Neumann és Goldstine elhagyták a Moore School-t, és így nem dolgoztak együtt Eckerttel és Mauchly-val, a Mérnöktovábbképző Intézetben egy új gép tervezésébe kezdtek, melyet IAS-gépnek neveznek. Neumannt számos egyetem, akadémia rendes és tiszteletbeli taggá választotta. 1952–54 között az USA Atomenergetikai Felügyelete Tanácsadó Bizottságának, 1953–57 között a Technikai Tanácsadó Testületnek tagja, 1955–57 között az USA Atomenergetikai kormánybiztosa. 1947-ben kapta meg a Medal for Merit, 1957-ben a Medal of Freedom kitüntetést, az Albert Einstein emlékérmet és az Enrico Fermi érdemrendet. 1957. február 8-án Washingtonban halt meg.

John Presper ECKERT (1919–)

1943-ban szerzett diplomát a pennsylvaniai Moore Schoolban. A diplomaszerzés után továbbra is a Moore Schoolban maradt, ahol John W. Mauchly-val együttműködve hadügyi megrendelésre hozzálátott a számítógépfejlesztéshez. A Moore Schoolban 1943 és 1946 között dolgoztak az ENIAC számítógépen. (ENIAC = Electrical Numerical Integrator and Computer.) Ez volt a világ legelső, működőképes numerikus elektronikus számítógépe. Az ENIAC létrehozása csapatmunka volt. Mégis első helyen Eckert állt, akinek hozzájárulása mindenki másét meghaladó mértékű volt. Főmérnökként ő volt a lelke az egész folyamatnak. Mauchly szerepe főleg abban állt, hogy a kezdeti ötleteket adta. Az ENIAC 18 ezer elektroncsövet és ezerötszáz jelfogót tartalmazott. Decimális gép volt, egy összeadás vagy kivonás átlagos ideje 200 μs volt. Egy szorzás 2 ms-ig tartott, az osztás pedig 30 ms-ig. Az adatokat egy szabványos IBM kártyaolvasóról vitték be a gépbe, s egy jelfogóból álló egységben tárolták. Az eredményeket kártyákra lyukasztották, majd a kártyákat egy IBM tabulátorral nyomtatták ki.

Az ENIAC programozása meglehetősen nehézkes volt, ezért nem véletlen, hogy csak egyetlen példány készült el belőle. Kezdetben katonai alkalmazásokra, ballisztikai táblázatok kiszámolására használták. 1944 végén Eckert azzal a gondolattal állt elő, hogy a késleltető művonalat föl lehetne használni információtárolásra. Különös szerencse volt, hogy amikor éppen ez az ötlet fölmerült, lépett színre Neumann János. Az ENIAC-ot követő gép, az EDVAC Neumann János logikai tervei alapján készült, és több vonatkozásban is lényeges előrelépést jelentett az ENIAC-hoz képest. (EDVAC = Electrical Discrete Variable Computer.) Az EDVAC létrehozásában döntő szerepet játszott Neumann, Eckert, Mauchly, Burks és Goldstine. 1944-ben létrehozták az Eckert–Mauchly Computer Corporation nevű céget. 1949-ben megszületett az új cég első üzleti célú számítógépe, a BINAC, ez azonban nem volt igazán sikeres.

Ezt követő modelljük, az UNIVAC (Universal Automatic Computer) annál nagyobb sikert aratott. 1950-ben a Remington Rand cég megvásárolta az Eckert–Mauchly céget, és

a UNIVAC I. gyártmánnyal jelentős piaci sikert aratott. Ezzel a Remington Rand megszerezte a technológiai vezető szerepet az éppen bontakozó *számítógép-ipar*ban. Előnye csak fokozódott 1955 közepén, amikor fúzióra léptek a Sperry Gyroscope-val és fölvették a Sperry Rand nevet. 1955-ben Eckert a Sperry Rand elnökhelyettese és kutatási igazgatója lett. 1957-ben kereskedelmi igazgató, 59-ben első elnökhelyettes, 63-ban pedig elnökhelyettes és elnöki tanácsadó lett, szintén a Sperry Rand cégnél. 1964-ben Eckertet megválasztották a Pennsylvania egyetem díszdoktorává. 1969-ben megkapta a legnagyobb amerikai tudományos kitüntetést, a National Medal of Science-t. (Lásd még Neumann J., H. H. Goldstine és J. W. Mauchly.)

Hermann H. GOLDSTINE (1913–)

Matematikus. 1936-ban doktorált a chicagói egyetemen, utána ugyanitt a matematikai intézetben dolgozott kutatói segédként. Gilbert A. Bliss, kiváló matematikus közeli munkatársaként matematikát tanított Bliss tanfolyamain. Ezután ballisztikát oktatott a michigani egyetemen. 1942-ben főhadnagyként katonai szolgálatra hívták, és a Ballisztikai Kutató Laboratórium állományába került. Feladata a philadelphiai egyetem Moore-intézetében fejlesztett differenciál analizátoron folyó ballisztikai számítások hatékonyságának fokozása volt. Ennek kapcsán vált szükségessé az első elektronikus digitális számítógép, az ENIAC megépítése. Kulcsszerepe volt az ENIAC-program elfogadtatásában. 1943-tól a Ballisztikai Kutató Laboratóriumot képviseli a projektben, maga is tevékenyen részt vesz elsősorban a programozásban és a kutatásban.

Neumann János Goldstine meghívására látogatta meg 1944 nyarán az ENIAC-ot, a látogatás eredményeként Neumann novemberben csatlakozott a számítógép fejlesztőihez. Ekkor kezdődött Neuman és Goldstine „hosszú és rendkívül gyümölcsöző baráti és munkakapcsolata, melynek csak Neumann korai halála vetett véget". Az ENIAC-ot követően új elvű gép, az EDVAC előzetes tervét Neumann már Goldstine közreműködésével dolgozta ki 1944 júliusában. 1945-ben szakítanak Eckerttel és Mauchlyval, Goldstine 1946-ban leszerel, és a Mérnöki Továbbképző Intézetbe megy Neumannhoz. Együtt tervezik és építik meg az úgynevezett IAS gépet, ami a világon az első igazi Neumann-elvű gép. Hosszú ideig dolgozott az IBM-nél, mint az Amerikai Filozófiai Társaság ügyvezető igazgatója.

Az ENIAC programban Adele Goldstine, Hermann Goldstine felesége is részt vett, ő volt a gép első hivatásos programozója. Tevékenyen részt vett a továbbfejlesztett ENIAC utasításrendszerének kidolgozásában és az első programozási kézikönyv megírásában. (Lásd még Neumann J., J. P. Eckert és J. W. Mauchly.)

John William MAUCHLY (1907–)

A jelenlegi elektronikus digitális számítógépes korszak egyik legjelentősebb úttörője. Szellemi terméke, az ENIAC 1946-ban teljesen megváltoztatta a tudományos és kereskedelmi információfeldolgozás addigi módszereit. 1925-ben szerzett mérnöki diplomát a John Hopkins egyetemen. Két év után úgy döntött, hogy mégsem lesz mérnök, hanem fizikával akar foglalkozni. 1932-ben doktori címet szerzett a *szénmonoxid-molekulák analízise* témában. A John Hopkins egyetemen fizikusként dolgozott, és többek között a formaldehid spektrum energiaszintjeinek számításával foglalkozott.

Ez a munka óriási mennyiségű számolási feladattal járt. Ekkor kezdett érdeklődni a különböző, gyors számítási eljárások iránt. 1933 és 1945 között az Ursinus College-ban fizikát tanított. Ez idő alatt az időjárás-előrejelzés érdekelte, és épített egy analóg számítógépet az időjárási adatok harmonikus analízisére. 1941 nyarán a fenyegető háborúra tekintettel szakértőként részt vett egy katonai védelmi elektronikai kurzuson, a Moore School-ban. A Moore School akkor már szerződéses kapcsolatban állt a hadsereg légelhárításával és ballisztikus táblázatokat készítettek. Mauchly ezekben a munkákban is részt vett, rendszeres oktatási feladatainak ellátása mellett. 1941-ben találkozott Atanasoff-fal, akinek sok ötletét később hasznosította az ENIAC építésekor. 1942 és 1946 között a Moore Schoolban Presper J. Eckerttel és Hermann Goldstine-nal együtt megépítette az ENIAC-ot. 1946-ban Eckerttel együtt elhagyta a Moore Schoolt. Nem sokkal később megalakították az Eckert—Mauchly Computer Corporationt. 1951-ben ez a vállalat beolvadt a Remington Randba. 1959-ben Mauchly Associates néven önálló vállalatot alakított. (Lásd még J. Presper Eckert és H. H. Goldstine.)

Alain M. TURING (1912—1954)

Angol matematikus. Egészen fiatalon kitűnik matematikai tehetségével. 1935-ben Cambridge-ben diplomát szerzett. Ezután kezdett matematikai logikával foglalkozni. 1937-ben, mindössze 25 éves korában jelentette meg híres írását: "On Computable Numbers with an Application to the Entscheidungsproblem". Ebben vezette be a később Turing-gépnek nevezett számítógép elvi alapjait. Tanulmánya hatására meghívták Princetonba, ahol 1938-ban doktori címet szerzett. Rövid ideig Neumann János munkatársa volt, de hamarosan hazatért hazájába, Angliába, Cambridge-be. A második világháború alatt katonai szolgálatban tudományos kutatásra sorozták be. 1939-től 45-ig volt a brit külügyminisztériumban. Közben alapos impulzustechnikai jártasságra tett szert, és ez döntően befolyásolta későbbi pályáját. 1942-ben hivatalos minőségben az Egyesült Államokban járt, ahol láthatta a legújabb számítógépes kutatásokat, és felújíthatta régi princetoni ismeretségeit. 1945-ben NPL (National Physical Laboratory) újonnan alakult matematikai osztályán kapott állást. Minden energiájával elektronikus számítógép építésébe vetette magát. Gépét ACE-nek (Automatic Computing Engine*(automatikus számoló motornak)* nevezte.

Bár tudott valamennyit Neumann gépéről, az EDVAC-ról, mégis teljesen külön úton járt, Babbage nyomdokait követte. Az ACE Turing megfogalmazásában „túlságosan ambíciózus gép volt kora elektronikai technológiájához képest". 1950-ben készült el a Turing gép, az ACE, amely nagy sikert aratott, és harminc változatban gyártotta az English Electric Company DEUCE név alatt. 41 éves korában, 1954-ben tragikus hirtelenséggel hunyt el.

John ATANASOFF (1903—)

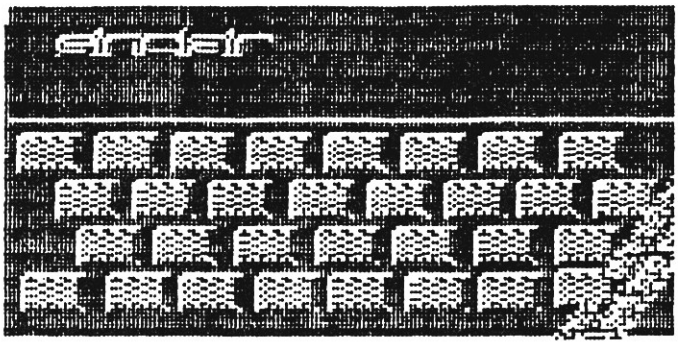
John Atanasoff a New York állambeli Hamiltonban született. 1926-ban szerzett diplomát az Iowa State Universityn, majd 1930-ban Wisconsinban matematikai fizikából doktorált. 1930 és 1945 között az Iowa State University-n volt matematika professzor. Számoló kapacitásának növelése érdekében már a 30-as évek legelején asztali számológépet akart készíteni saját használatra. Elhagyta az analóg eszközöket, mert túlságosan lassúnak és nem eléggé pontosnak találta azokat, és a digitális technikára koncentrált.

1957-re vált világossá számára, hogy a számítások gépesítésének teljes útját az elektronikus digitális eszközök jelentik.

Egy Clifford Berry nevű diákjával együtt 1939-ben készült el számítógépének prototípusával. Végül is 1942-re készült el teljesen az ABC (Atanasoff—Berry Computer) számítógép. Ez lineáris egyenletrendszerek megoldására szolgáló elektronikus gép volt, és rengeteg új ötletet tartalmazott. Az ABC gép logikai áramköreit, felfrissíthető memóriáját, forgódobos kondenzátoros tárolómágnesét sok későbbi gépben láthatjuk viszont. Atanasoff ekkor még nem érezte meg a számítógép előtt álló óriási lehetőségeket, és azzal sem volt tisztában, milyen óriási mértékben járult hozzá a számítógépes technológia kifejlesztéséhez.

Nem is sejtette, hogy gépe milyen fontos a számítógépes technológia kialakulása szempontjából. Ezért ötleteit nem is szabadalmaztatta. Egész életében matematika professzor maradt, és ma, 84 éves korában is matematikával foglalkozik. Csak az 1970-es években fedezték fel Atanasoffot, mint a digitális számítástechnika egyik atyját. 1973-ban a Honeywell jogi elégtételt szerzett neki azzal, hogy szabadalmi pert indított John Mauchly és J. Presper Eckert ellen. A pert megnyerték, vagyis a bíróság kimondta, hogy Mauchly és Eckert nagyon sok ötletet vett át Atanasofftól, a világ első elektronikus számítógépének, az ENIAC-nak megépítéséhez. A bíróság megállapította, hogy Mauchly 1941-ben az Iowa State Universityn Atanasoffnál járt, majd ezután 1942 és 1946 között a Pennsylvania Egyetem Moore Intézetében felépített ENIAC számítógépéhez számtalan ötletet vettek át Atanasoff-tól, és azt saját nevükön szabadalmaztatták. A bíróság a szabadalmak nagy részét Atanasoff nevére íratta át, de ennek csak erkölcsi jelentősége van, hiszen az évtizedekkel ezelőtt felvett szabadalmi jogdíjak már nemigen perelhetők vissza.

Sir Clive SINCLAIR



A mikroszámítógépek néhány éves történelmének egyik zsenije. Szakmai sikerekkel és bukásokkal a háta mögött újabb próbára készül: a **Z 88** telepes, hordozható, folyadékkristály-kijelzős, 8 bites gépe már megjelent a házi számítógépek piacán. A NAGY NEVEK közé e kötet karikaturistájának különös szimpátiája révén került: Sir Clive *Spectrum*a nélkül ugyanis aligha illusztrálhatta volna könyvünket.

John KEMÉNY és Thomas KURTZ

A Basic programozási nyelv kifejlesztői. (Basic=Beginner's All-purpose Symbolic Instruction Code.) A Basic programozási nyelv a Tarthmouth College-ban jött létre, az 1960-as évek közepén. Könnyen tanítható, olcsón alkalmazható programozási nyelv. Kifejezetten oktatási célra fejlesztették ki, mivel nagy tömegek számára könnyen elsajátítható, interaktív nyelv, vagyis a programozó azonnal választ kap.

KOZMA László (1902–1983)

Mérnöki diplomáját Brünnben szerezte, 1930-tól a Bell Telephone antwerpeni gyárában telefonközpont-fejlesztőként dolgozott. 1936-ban Belgiumban megbízták, hogy tervezzen és építsen telefonközpont-elemekből automata számológépet. 1938-ban készítette el az első jelfogós számítógépét, amelyre angol és USA-szabadalmat is kapott. Ezt a gépet további kettő követte, és még számos találmány, mint pl. egy távfeldolgozó rendszer telex-terminálokkal. A háború után hazajött Magyarországra, és 1955-ben megépítette a Budapesti Műszaki Egyetemen az első magyar gyártmányú jelfogós számítógépet, a MESZ I.-et.

A MESZ I. bináris számítógép volt, programvezérelt, de a szó mai értelmében nem tárolt programú gép, ugyanis a programot kézzel lyukasztott lyukkártya tartalmazta. A berendezés mintegy kétezer darab jelfogóból épült, az eredmény kiírására egy írógépet alakítottak át. Ez a gép mintegy tíz évig működött. Kozma László akadémikus, a Műegyetem professzora; élete végéig főként a hazai telefonközpont-rendszer fejlesztésén dolgozott.

KALMÁR László (1905–1976)

A magyarországi számítástechnika úttörője volt, akadémikus, a József Attila Tudományegyetem professzora. Többek között formális logikát adott elő. Szegedi, vagy Kalmár-féle logikai gépét 1960-ban mutatta be, munkatársával Muszka Dániellel. A gépet háromvezetékes huzalrendszerrel lehetett programozni. Jelfogókból és számjegygépekből összeszerelt vezérlőmű vizsgálta meg a programozott logikai feladat minden egyes variációját, és megállapította, hogy a kívánt bonyolult ítéletsorokból álló összetett ítélet milyen feltételek mellett igaz, vagy hamis. Korai halála akadályozta meg egy új „formulavezérlésű" gépe terveinek befejezését, mely az emberi kommunikációhoz közel álló módon lett volna programozható.

Sokáig szinte egy emberként harcolt a Tudományos Akadémián e tudományág hazai elismeréséért. A JATE gyakorlatilag 1960 óta a programtervező matematikusok képzésének egyik központja. Kalmár mint tudományszervező érdemei is elévülhetetlenek. Számítógépeseink egész nemzedéke büszke arra, hogy Kalmár-tanítvány.

69.

EPILÓGUS HELYETT
(Búcsú az olvasótól)

Vannak dolgok, amelyek maguktól véget érnek, és vannak, amelyeket abba kell hagyni. Engedje meg az olvasó, hogy könyvünket ebből a szempontból a második kategóriába soroljuk. És ha a többi fejezet nem is volt mindig kellően komoly (mármint ami a számítástechnikát illeti), legalább ez az utolsó legyen az. Tehát:

STOP READ

CLOSE FILES

GOODBYE!

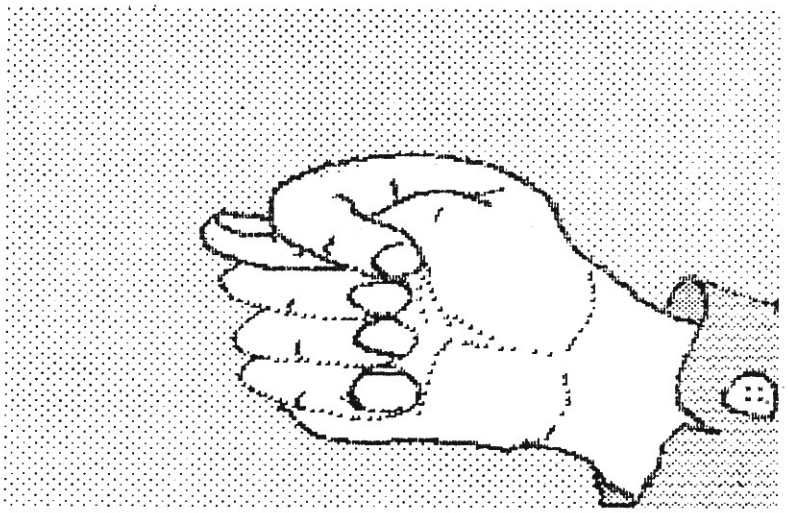
SZÁMÍTÁSTECHNIKAI HOL–MI

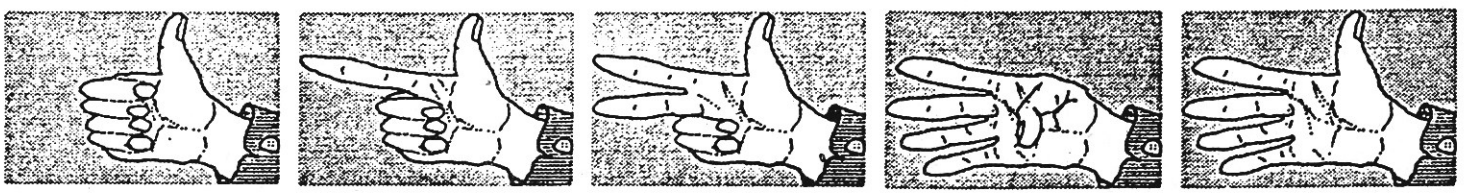
(Tárgymutató)

és

SZÁMITÁSTECHNIKAI MI–MI

(Szótárka)





HOL–MI
(Tárgymutató)

ábrázolási mód 15, 16
adatátviteli hálózat *lásd: hálózat*
adatbank 57
adatbázis 57, 58
adatbáziskezelő rendszer 57
adatbevitel 4, 45
adatbiztonság 39, 40, 52, 65
adatmegjelenítő 19
adatregiszter 12
adatvédelem *lásd: adatbiztonság*
adatrögzítő 7
aktív memóriakártya 62
alapkártya 18
alkalmazási programozó 7
aritmetikai utasítás 14
aritmetikai–logikai egység 2, 9, 12
ASCII kód 15
bájt 10, 12
beviteli-kiviteli vezérlőegység 9
billentyűzet 18, 19
bit 10
bináris aritmetika 11
bináris számábrázolás 16
biztonság *lásd: adatbiztonság*
CD *lásd: kompakt diszk*
ciklusidő 12
cím 10, 12, 14
címregiszter 12
címrész (utasítás) 12
csillaghálózat 27
dBASE 57
decimális számábrázolás 16
dokumentum formátumú adatbázis 58
dokumentum-visszakereső rendszerek 58
editor 55
EAN/VPC kód 25
EBCDIC kód 15
ellenőrző bájt 17
elektronikus pénz 62
elektronikus posta 59
elérési idő 12
ENIAC 3
eredményszolgáltatás 4
feldolgozás 4
felhasználó 7
félvezető tár 13
file 43, 44
fix lemez *lásd: winchester-tár*
floppy (flopi) *lásd: hajlékonylemez*
folyamatszervező 7
fordítóprogramok 32
GABRIEL helyfoglalási rendszer 28
gazdaságosság 64
generáció 3
géphasználati díj 64
gépi kód 11, 12, 14
gyűrűs hálózat 27
hajlékonylemez 23
hálózat 2, 3, 27, 28
hardver 2
háttértároló *lásd: tömegtároló*
házi személyi számítógép 5
hibakeresés (program) 48
hibavédelem (hardver) 17
hierarchikus hálózat 27
hozzáférés 22
IDMS rendszer 57
időosztás 27
input *lásd: adatbevitel*
input–output processzor 9, 12
intelligens terminál 27
irodaautomatizálás 59
jelátalakító 28
karbantartó 7
képernyős terminál *lásd: adatmegjelenítő*
kerekítés 46
kilobájt 10
kisszámítógép 5
kitüntetett utasítás 14
kód 15

kódolási kulcs 63
kódoló 7
kompakt diszk 26
költségek (feldolgozási) 64
központi egység 9, 12
központi tár *lásd: operatív tár*
közvetlen hozzáférés 22
lebegőpontos számábrázolás 16
lézernyomtató 21
logikai utasítás 14
margarétakerekes nyomtató 20
magyar ábécé 56
mágneskazetta 22
mágneslemez 22, 23
mágnesszalag 22
mátrixnyomtató 20
MEDIAC rendszer 60
megabájt 10
memória 9, 12, 15
memóriakártya 62
mesterséges intelligencia 3, 60, 66
mikrogép *lásd: személyi számítógép*
miniszámítógép *lásd: kisszámítógép*
modell 61
modem *lásd: jelátalakító*
multiprogramozás 33, 34
Murphy törvényei 8, 29, 51, 54
műveleti rész (utasítás) 12
nagyszámítógép 5, 53
nyomtató 20, 21
OCR jelszabvány 24
operációs rendszerek 31, 50
operatív tár 2, 22, 9
optikai adattárolás 26
optikai bizonylatolvasó 24
orvosi alkalmazás 60, 66
output *lásd: eredményszolgáltatás*
ötödik generációs gépek 3, 7
paritás 17
PC *lásd: személyi számítógép*
periféria 2, 12
processzor 2, 9, 12, 18
professzionális személyi számítógép 5
programtervező 7
RAM 13
rendszerprogramozó 7
rendszerszervező 7
regiszter 12
rejtjelezés 63
repülésszimulátor 61
ROM 13
sornyomtató 20
soros editor 55
soros hozzáférés 22
szakértői rendszerek 60, 66
számábrázolás 16
számítástechnikai munkakörök 7, 52
számítógépes hálózat *lásd: hálózat*
személyi számítógép 5
szimbolikus cím 12
szimuláció 61
szinonimaszótár 57
szó 12
szoftver 2, 36, 37
szoftver értéke 37
szoftver védelme 38
szolgáltató programok 36
szövegesinformáció-visszakereső rendszer 57
szövegszerkesztők 55
szupernagy-számítógép 5
tabló 20
teljes képernyős szerkesztő 55
terminál 19, 27
titkosság 65
Toshiba levélosztályozó 24
többfunkciós kártya 18
tömegtároló 2, 9, 22, 23
utasításkód 14
üzemeltető személyzet 7
valós idejű rendszer 27, 28
változónév 12
vezérlésátadó utasítás 14
vezérlőmű 2, 9, 12
vonalkód 25
winchester-tár 23

A kifejezések után álló számok azokra a fejezetekre utalnak, ahol az adott témát részletesebben ismertettük.

MI–MI
(Szótárka)

MAGYAR–ANGOL

ábécé	ABC	*é-bí-szí*
ablak	window	*vindó*
ábrázolás	representation	*reprezentésn*
ábrázolás	notation	*notésn*
adat	data	*déta*
adatállomány	file	*fájl*
adatátvitel	data transmission	*déta trenzmisn*
adatbank	data bank	*déta benk*
adatbázis	data base	*déta bész*
adatellenőrzés	data control	*déta kontrol*
adatfeldolgozás	data processing	*déta prószeszing*
adathordozó	data medium	*déta médium*
adatrögzítés	data recording	*déta rikording*
adatvédelem	data protection	*déta proteksn*
akkumulátor	accumulator	*ekjumulétor*
alacsony szintű nyelv	low level language	*lo le-vl lengvidzs*
alácsordulás	underflow	*anderfló*
alapértelmezés	default	*difólt*
alfabetikus	alphabetic	*alfabetik*
alfanumerikus	alphanumeric	*alfanjumerik*
algoritmus	algorithm	*elgoritm*
állandó	constant	*konsztent*
alprogram	subroutine	*szabrutin*
ALU (Aritmetikai és logikai egység)	ALU (arithmetic and logic unit)	*é-el-jú (eriszmetik end lodzsik junit)*
analóg	analog	*enelog*
argumentum	argument	*árgjument*
aritmetika	arithmetic	*eriszmetik*
ASCII-kód (Amerikai szabványos információcsere-kód)	ASCII (American Standard Code for Information Interchange)	*eszki (emeriken sztender kód for informesen intercséndzs)*
assembly nyelv	assembly language	*eszembli lengvidzs*
asszembler	assembler	*eszembler*
áthozat	borrow	*boro*
átvitel	carry	*keri*
automatikus	automatic	*ótometik*
azonos idejű	real-time	*riel-tájm*
azonosító	identifier	*ájdentifájer*
bájt	byte	*bájt*
Basic	BASIC	*bészik*
be-kiviteli (I/O)	input/output (I/O)	*input/autput (aj/o)*
bemenet	input	*input*
bemutató	demo	*demó*
beolvasás	read	*ríd*
betöltés	loading	*lóding*
betű	character	*karakter*
betűváltó	shift	*sift*
billentyű	key	*kíj*
billentyűzet	keyboard	*kíjbórd*
bináris szám	binary digit	*bájneri didzsit*

ANGOL–MAGYAR

ABC	é-bí-szí	*ábécé*
access	ekszesz	*elérés(i)*
accumulator	ekjumulétor	*akkumulátor*
addition	edisn	*összeadás*
address	edresz	*cím*
algorithm	elgoritm	*algoritmus*
alphabetic	alfabetik	*alfabetikus*
alphanumeric	alfanjumerik	*alfanumerikus*
ALU (arithmetic and logic unit)	é-el-jú (eriszmetik end lodzsik junit)	*ALU (aritmetikai és logikai egység)*
analog	enelog	*analóg*
argument	argjument	*argumentum*
arithmetic	eriszmetik	*aritmetika*
array	eréj	*mező*
artifical intelligence	artifikl intellidzsensz	*mesterséges intelligencia*
ASCII (American Standard Code for Information Interchange)	eszki (emeriken sztenderd kód for informesen intercséndzs)	*ASCII-kód*
assembler	eszembler	*asszembler*
assembly language	eszembli lengvidzs	*assembly nyelv*
automatic	ótometik	*automatikus*
BASIC	bészik	*Basic*
batch	becs	*köteg(elt)*
binary	bájneri	*kettes (számrendszer)*
binary digit	bájneri didzsit	*bináris szám*
bit	bit	*bit*
block diagram	blok dajegrem	*blokkdiagram*
Boolean operation	búlien operésn	*Boole-művelet*
borrow	boró	*áthozat*
branch	bráncs	*elágaz(tat)ás*
break point	brék pojnt	*megszakítási pont*
buffer store	bafersztór	*puffertár*
bus	basz	*sín*
byte	bájt	*bájt*
C language	szi lengvidzs	*C nyelv*
CAI (Computer Aided Instruction)	szí-é-áj (kompjuter ejdid insztraksn)	*CAI (számítógéppel segített oktatás)*
calling	kóling	*hívás*
card	kárd	*kártya*
carry	keri	*átvitel*
cassette	keszet	*kazetta*
catalogue	ketelog	*katalógus*
channel	csenel	*csatorna*
character	karakter	*betű*
character	karakter	*karakter*
check	csek	*ellenőrzés(i)*
chip	csip	*lapka*
clock signal	klok szignel	*órajel*
code	kod	*kód*

bit	bit	*bit*
bizonylat	document	*dokjument*
blokkdiagram	block diagram	*blok dajegrem*
bogarászás	debugging	*dibaging*
Boole-művelet	Boolean operation	*búlien operésn*
botkormány	joystick	*dzsojsztik*
C nyelv	C language	*szí lengvidzs*
CAI (számítógéppel segített oktatás)	CAI (Computer Aided Instruction)	*szí-é-áj kompjuter éjdid insztraksn*
ciklus	cycle	*szajkl*
cím	address	*edresz*
címke	label	*lébl*
csak olvasható tár	ROM (read only memory)	*rom (ríd onli memori)*
csatorna	channel	*csenel*
decimális	decimal	*deszimel*
definíció	definition	*definisn*
deklaráció	declaration	*deklarésn*
demolemez	demodisc	*demódiszk*
digitális	digital	*didzsital*
digitalizáló	digitizer	*didzstájzer*
dinamikus táblázat-kezelő program	spreadsheet	*szpred-sít*
direkt	direct	*dájrekt*
diszkrét	discrete	*diszkrít*
dokumentáció	dokumentation	*dokjumentésn*
dupla	double	*dabl*
egér	mouse	*mausz*
egész	integer	*intedzser*
egység	unit	*junit*
elágaz(tat)ás	branch	*bráncs*
elektronika	electronics	*ilektroniksz*
elérés(i)	access	*ekszesz*
eljárás	procedure	*proszidzsör*
ellenőrzés	verify; verification	*verifáj; verifikésn*
ellenőrzés(i)	check	*csek*
előjel	sign	*szájn*
emulátor	emulator	*emuljétor*
EPROM	EPROM	*iprom*
értelmezőprogram	interpreter	*interpréter*
fej	head	*hed*
felhasználó	user	*júzer*
felhasználó-barátságos	user friendly	*júzer frendli*
felhasználói kézikönyv	user guide	*júzer gájd*
feltétel	condition	*kondisn*
feltételes	conditional	*kondisnel*
felügyelőprogram	supervisor	*szupervájzor*
félvezető	semiconductor	*szemikondaktor*
fényceruza	light pen	*lájt pen*
folyamatábra	flow chart	*fló csart*
fordító (program)	compiler	*kompájler*
formázás	formatting	*formetting*
forrás	source	*szórsz*
függvény	function	*fanksn*
fut(tat)ás	running	*ráning*
füzér	string	*sztring*
generáció	generation	*dzsenerésn*
gépi kód	machine code	*mésin kód*
grafika	graphics	*grefiksz*
hajlékonylemez	floppy disk	*flopi diszk*
hálózat	net	*net*
hardver	hardware	*hárdver*
házi számítógép	home computer	*hóm kompjúter*

compatibility	kompetibiliti	*kompatibilitás*
compiler	kompájler	*fordító (program)*
computer	kompjúter	*számítógép*
condition	kondisn	*feltétel*
conditional	kondisnel	*feltételes*
configuration	konfigurésn	*konfiguráció*
constant	konsztent	*állandó*
control	kontrol	*vezérlés*
copy	kopi	*másolás*
cursor	körszor	*kurzor*
cybernetics	szájbernetiksz	*kibernetika*
cycle	szajkl	*ciklus*
daisywheel	déjzivil	*margarétakerék*
data	déta	*adat*
data bank	déta benk	*adatbank*
data base	déta bész	*adatbázis*
data control	déta kontrol	*adatellenőrzés*
data medium	déta médium	*adathordozó*
data processing	déta prószeszing	*adatfeldolgozás*
data protection	déta proteksn	*adatvédelem*
data recording	déta rikording	*adatrögzítés*
data transmission	déta trenzmisn	*adatátvitel*
debugging	dibaging	*bogarászás*
debugging	dibaging	*hibakeresés*
decimal	deszimel	*decimális*
declaration	deklarésn	*deklaráció*
default	difólt	*alapértelmezés*
definition	definisn	*definíció*
demo	demó	*bemutató*
demodisc	demódiszk	*demolemez*
digital	didzsital	*digitális*
digitizer	didzstájzer	*digitalizáló*
direct	dajrekt	*direkt*
direct	direkt	*közvetlen*
discrete	diszkrít	*diszkrét*
disk	diszk	*lemez*
display	diszpléj	*kijelző*
display	diszpléj	*megjelenítő*
division	divizsn	*osztás*
document	dokjument	*bizonylat*
documentation	dokjumentésn	*dokumentáció*
dot (matrix) printer	dot (métriksz) printer	*mátrixnyomtató, mozaiknyomtató*
double	dabl	*dupla*
driver	drájver	*meghajtó*
dummy	dámi	*üres*
electronics	ilektroniksz	*elektronika*
emulator	emuljétor	*emulátor*
EPROM	iprom	*EPROM*
erase	iréz	*törlés*
error correction	eror koreksn	*hibajavítás*
execution	ekszikjusn tájm	*végrehajtás*
expression	ikszpresn	*kifejezés*
file	fájl	*adatállomány*
floating point	flőting pojnt	*lebegőpont*
floppy disk	flopi diszk	*hajlékonylemez*
flow chart	fló csart	*folyamatábra*
formatting	fórmetting	*formázás*
function	fanksn	*függvény*
game program(me)	gém progrem	*játékprogram*
gate	gét	*kapu*
generation	dzsenerésn	*generáció*
graphics	grefiksz	*grafika*

helyi hálózat	local network	*lokel netvörk*
hibajavítás	error correction	*eror koreksn*
hibakeresés	debugging	*díbáging*
hívás	calling	*cóling*
hőnyomtató	thermal printer	*törmel printer*
IBM	IBM	*áj-bí-em*
időosztás	time-sharing	*tájm-sering*
illesztő	interface	*interfész*
indexszekvenciális	index sequential	*indeksz szikvensl*
információ	information	*informésn*
informatika	informatics	*informatiksz*
inicializálás	initialization	*iniselájzésn*
input	input	*input*
intergált áramkör	integrated circuit	*integrétid szörkit*
interaktív	interactive	*interektiv*
inverz	inverse	*invörsz*
író-olvasó fej	read-write head	*ríd-rájt hed*
játékprogram	game program(me)	*gém progrem*
jel	signal	*szignel*
jelkép	symbol	*szimból*
kapcsoló	switch	*szvics*
kapu	gate	*gét*
karakter	character	*karakter*
kártya	card	*kárd*
katalógus	catalogue	*ketelog*
kazetta	cassette	*keszet*
kép	picture	*pikcsör*
képernyő	screen	*szkrín*
képpont	pixel	*pikszel*
keresés	searching	*szörcsing*
kettes (számrendszer)	binary	*bájneri*
kibernetika	cybernetics	*szájbernetiksz*
kifejezés	expression	*ikszpresn*
kijelző	display	*diszpléj*
kilobájt	kilobyte	*kilobájt*
kimenet	output	*autput*
klaviatúra	keyboard	*kíjbórd*
kód	code	*kód*
kompatibilitás	compatibility	*kompetibiliti*
konfiguráció	configuration	*konfigurésn*
konstans	constant	*konsztent*
könyvtár	library	*lájbreri*
köteg(elt)	batch	*becs*
közvetlen	direct	*direkt*
kulcsszó	keyword	*kíjvörd*
kurzor	cursor	*körszor*
lap	page	*pédzs*
lapka	chip	*csip*
lebegőpont	floating point	*flőting pojnt*
lemez	disk	*diszk*
lézernyomtató	laser printer	*lézerprinter*
lista	list	*liszt*
logikai	logical	*lodzsikel*
lyukkártya	punched card	*puncst kard*
MACINTOSH	MACINTOSH	*mekintos*
mag	nucleus	*nyúkliösz*
magas szintű	high level	*haj level*
mágnes(es)	magnetic	*magnetik*
makro	macro	*mekró*
margarétakerék	daisywheel	*déjzivil*
másolás	copy	*köpi*
mátrixnyomtató	dot (matrix)	*dot (métriksz)*
mozaiknyomtató	printer	*printer*

hardware	hárdver	*hardver*
head	hed	*fej*
high level	haj level	*magas szintű*
home computer	hóm kompjúter	*házi számítógép*
IBM (International Business Machine)	áj-bí-em	*IBM*
identifier	ájdentifájer	*azonosító*
index sequential	indeksz szikvensl	*indexszekvenciális*
informatics	informatiksz	*informatika*
information	informésn	*információ*
initialization	iniselájzesn	*inicializálás*
input	input	*bemenet*
input	input	*input*
input/output (I/O)	input/autput (aj/o)	*be-kiviteli (I/O)*
instruction	insztraksn	*utasítás*
integer	intedzser	*egész*
integrated circuit	integrétid szörkit	*integrált áramkör*
interactive	interektiv	*interaktív*
interface	interfész	*illesztő*
interpreter	interpréter	*értelmezőprogram*
interrupt	interapt	*megszakítás*
inverse	invörsz	*inverz*
job	dzsob	*munka*
joystick	dzsojsztik	*botkormány*
jump	dzsamp	*ugrás*
key	kíj	*billentyű*
keyboard	kíjbórd	*billentyűzet*
keyboard	kíjbórd	*klaviatúra*
keyword	kíjvörd	*kulcsszó*
kilobyte	kilobájt	*kilobájt*
label	lébl	*címke*
language	lengvidzs	*nyelv*
laser printer	lézerprinter	*lézernyomtató*
library	lájbreri	*könyvtár*
light pen	lájt pen	*fényceruza*
line printer	lájn printer	*sornyomtató*
list	liszt	*lista*
loading	lóding	*betöltés*
local network	lokel netvörk	*helyi hálózat*
logical	lodzsikel	*logikai*
loop	lup	*ciklus*
low level language	lo le-vl lengvidzs	*alacsony szintű nyelv*
machine code	mésin kód	*gépi kód*
MACINTOSH	mekintos	*MACINTOSH*
macro	mekró	*makró*
magnetic	magnetik	*mágnes(es)*
memory	memori	*memória*
memory capacity	memori kepesziti	*tárkapacitás*
menü	menjú	*menü*
message	meszidzs	*üzenet*
microcomputer	májkrokompjúter	*mikroszámítógép*
microprocessor	májkroprószeszor	*mikroprocesszor*
mnemonic	nimonik	*mnemonika*
mouse	mausz	*egér*
multiprogramming	maltiprogreming	*multiprogramozás*
net	net	*hálózat*
notation	notésn	*ábrázolás*
nucleus	nyúkliösz	*mag*
number representation	namber reprezentésn	*számábrázolás*
numeric(al)	nyumerik(el)	*numerikus*
object language	obdzsekt lengvidzs	*tárgy nyelv*
offline	oflájn	*offline*
online	onlájn	*online*

megbízhatóság	realibility	*rilájebiliti*
meghajtó	driver	*drájver*
megjegyzés	remark	*rimárk*
megjelenítő	display	*diszpléj*
megszakítás	interrupt	*interapt*
megszakítási pont	break point	*brék pojnt*
memória	memory	*memori*
menü	menu	*menjú*
mesterséges intelligencia	artifical intelligence	*artifikl intellidzsensz*
mező	array	*eréj*
mikroprocesszor	microprocessor	*májkroprócesszor*
mikroszámítógép	microcomputer	*májkrokompjúter*
mnemonika	mnemonic	*nimonik*
multiprogramozás	multiprogramming	*maltiprogreming*
munka	job	*dzsob*
mutató	pointer	*pojnter*
művelet	operation	*operésn*
numerikus	numeric(al)	*nyúmerik(el)*
nyelv	language	*lengvidzs*
nyomkövetés	trace	*tresz*
nyomtató	printer	*printer*
offline	offline	*oflájn*
online	online	*onlájn*
operációs rendszer	operating system	*operéting szisztem*
operatív tár	operating memory	*opereting memori*
operátor	operator	*operétor*
optikai leolvasó	optical scanner	*optikel szkenner*
órajel	clock signal	*klok szignel*
osztás	division	*divizsn*
összeadás	addition	*edisn*
papírtovábbító	tractor	*trektor*
paraméter	parameter	*peremíter*
parancs	statement	*sztétment*
paritás	parity	*periti*
párosság	parity	*periti*
periféria	periphery	*periferi*
PL/1	PL/1	*pí-el van*
pozíció	position	*pozisn*
processzor	processzor	*prószeszor*
program	program(me)	*progrem*
programozás	programming	*progreming*
prompt	prompt	*prompt*
puffertár	buffer store	*bafersztór*
rajzgép	plotter	*ploter*
RAM (közvetlen elérésű tár)	RAM (Random Access Memory)	*rem (rendom ekszesz memori)*
regiszter	register	*redzsiszter*
rekord	record	*rekord*
reláció	relation	*rilésn*
rendezés	sort	*szort*
rendszer	system	*szisztem*
ROM (csak olvasható tár)	ROM (Read Only Memory)	*rom (ríd onli memori)*
rutin	routine	*rutín*
sáv	track	*trek*
sín	bus	*basz*
sornyomtató	line printer	*lájn printer*
soros	serial	*szíriel*
streamer	streamer	*sztrímer*
szalag	tape	*tép*
számábrázolás	number representation	*namber reprezentésn*

operating memory	operéting memori	*operatív tár*
operating system	operéting szisztem	*operációs rendszer*
operation	operésn	*művelet*
operator	operétor	*operátor*
optical scanner	optikel szkenner	*optikai leolvasó*
output	autput	*kimenet*
overflow	óverfló	*túlcsordulás*
page	pédzs	*lap*
parameter	peremíter	*paraméter*
parity	periti	*paritás*
parity	periti	*párosság*
periphery	periferi	*periféria*
picture	pikcsör	*kép*
pixel	pikszel	*képpont*
PL/1	pi-el/van	*PL/1*
plotter	plóter	*rajzgép*
pointer	pojnter	*mutató*
position	pozisn	*pozíció*
printer	printer	*nyomtató*
procedure	proszidzsör	*eljárás*
processor	prószeszor	*processzor*
processor	prószeszor	*vezérlőegység*
program(me)	progrem	*program*
programming	progreming	*programozás*
prompt	prompt	*prompt*
punched card	puncst kard	*lyukkártya*
RAM (Random Access Memory)	rem (rendom ekszesz memori)	*RAM (közvetlen elérésű tár)*
random access	rendom ekszesz	*véletlen elérés*
read	ríd	*beolvasás*
read-write head	ríd-rájt hed	*író-olvasó fej*
real	ríel	*valós*
real-time	riel-tájm	*azonos idejű*
real-time	riel-tájm	*valós idejű*
record	rekord	*rekord*
register	redzsiszter	*regiszter*
relation	rilesn	*reláció*
realibility	rilajébiliti	*megbízhatóság*
remark	rimárk	*megjegyzés*
representation	reprezentésn	*ábrázolás*
response time	riszponz tájm	*válaszidő*
return	ritörn	*visszatérés*
ROM (Read Only Memory)	rom (ríd onli memori)	*ROM (csak olvasható tár)*
routine	rutín	*rutin*
running	ráning	*fut(tat)ás*
screen	szkrín	*képernyő*
searching	szörcsing	*keresés*
semiconductor	szemikondaktor	*félvezető*
serial	szíriel	*soros*
shift	sift	*betűváltó*
sign	szájn	*előjel*
signal	szignel	*jel*
software	szoftver	*szoftver*
sort	szort	*rendezés*
source	szórsz	*forrás*
space	szpész	*szóköz*
spreadsheet	szpred-sít	*dinamikus táblázat-*
sprite	szprájt	*szellem*
stack	sztek	*verem*
statement	sztétment	*parancs*
streamer	sztrímer	*streamer*

számítógép	computer	*kompjúter*
szellem	sprite	*szprájt*
szimbólum	symbol	*szimbol*
szó	word	*vörd*
szoftver	software	*szoftver*
szóköz	space	*szpész*
szövegfeldolgozás	word processing	*vörd prószeszing*
szövegszerkesztő	text editor	*tekszt editor*
szubrutin	subroutine	*szabrutin*
tárgy nyelv	object language	*obdzsekt lengvidzs*
tárkapacitás	memory capacity	*memori kepesziti*
távfeldolgozás	teleprocessing	*teleprószeszing*
telefax	telefax	*telefaksz*
terminál	terminal	*törminel*
törlés	erase	*iréz*
túlcsordulás	overflow	*óverfló*
ugrás	jump	*dzsamp*
utasítás	instruction	*insztraksn*
üres	dummy	*dámi*
ütemadó	timer	*tájmer*
üzenet	message	*meszidzs*
válaszidő	response time	*riszponz tájm*
valós	real	*ríel*
valós idejű	real-time	*riel-tájm*
változó	variable	*veriébl*
végrehajtás	execution	*ekszikjúsn*
véletlen elérés	random access	*rendom ekszesz*
verem	stack	*sztek*
vezérlés	control	*kontrol*
vezérlőegység	processzor	*prószeszor*
visszatérés	return	*ritörn*
winchester	winchester	*vincseszter*

string	sztring	*füzér*
subroutine	szabrutin	*alprogram*
subroutine	szabrutin	*szubrutin*
supervisor	szupervájzor	*felügyelőprogram*
switch	szvics	*kapcsoló*
symbol	szimból	*jelkép*
symbol	szimból	*szimbólum*
system	szisztem	*rendszer*
tape	tép	*szalag*
telefax	telefaksz	*telefax*
teleprocessing	teleprószeszing	*távfeldolgozás*
terminal	törminel	*terminál*
text editor	tekszt editor	*szövegszerkesztő*
thermal printer	törmel printer	*hőnyomtató*
time-sharing	tájm-sering	*időosztás*
timer	tájmer	*ütemadó*
trace	trész	*nyomkövetés*
track	trék	*sáv*
tractor	trektor	*papírtovábbító*
underflow	anderfló	*alácsordulás*
unit	junit	*egység*
user	júzer	*felhasználó*
user friendly	júzer frendli	*felhasználó-barátságos*
user guide	júzer gájd	*felhasználói kézikönyv*
variable	veriébl	*változó*
verify; verification	verifáj; verifikésn	*ellenőrzés*
winchester	vincseszter	*winchester*
window	vindó	*ablak*
word	vörd	*szó*
word processing	vörd prószeszing	*szövegfeldolgozás*

Minden szakmának megvan a maga szakzsargonja, és minden szakma irodalmának megvannak a nyelvészei is. Úgy gondoljuk, hogy az egységes szaknyelv kimunkálása nem mániákus nyelvészek öncélú törekvése, hiszen valamennyiünk érdeke, hogy egy szakmán belül egymást megértsük, a szakmára felkészülőket pedig tanítani tudjuk.

Szótárunkba a legfontosabb szavak, kifejezések közül válogattunk ki mintegy két és félszázat. Látható, hogy sok angol szakkifejezésnek nincs magyar megfelelője. Részben azért, mert a magyar szaknyelv átvette, magyarként használja azokat, részben pedig azért, mert a legügyesebb nyelvújító ötletek sem váltak általánosan elfogadottá. Ahol azonban magyar szavakat látunk, jegyezzük meg, használjuk bátran azokat.

A magyar kiejtés megadásánál kerültük a fonetkus jeleket, a magyar ábécé betűivel igyekeztünk az angol kiejtéshez közelálló, jó esetben annak megfelelő hangzást elérni. Az angol kiejtésnek ennél sokkal bonyolultabbak a szabályai, a kiejtési szótár szolgai használatának – sokszor – legalább olyan megmosolyogtató az eredménye, mint az angol szó „magyaros" olvasatának. Szótárunk tehát inkább csak támpont.

Ára: 125,–Ft

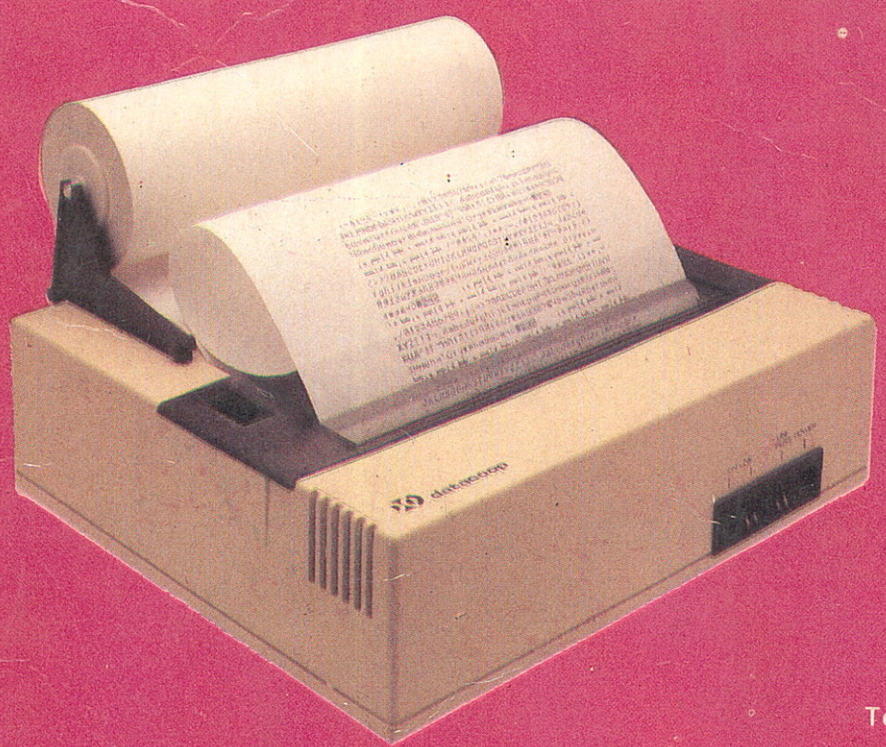
DCD-PRT-42 Babyprint

PROGRAMOZHATÓ MÁTRIX-NYOMTATÓ

SOKOLDALÚ FELHASZNÁLHATÓSÁG, KIS MÉRETEK, KEDVEZŐ ÁR!

Forgalmi adó mentes szakbolti ára: 19'500,–Ft
fogyasztói ára: 22'400,–Ft

A nyomtató többféle beépített interfésszel készül:
- a Commodore és VT személyi számítógéphez,
- a CENTRONIX interfészes alapváltozat saját csatlakozóval, kábel nélkül kapható,
- a PRIMO gépekhez a CENTRONIX változat illeszthető, (raktárról kapható) külön interfésszel.



A gyártó
– kívánságra –
más gépekhez való
illesztést is vállal.

További felvilágosítást
nyújtanak Önnek
az ELEKTROMODUL
szakboltjában:

Budapest XIII.
Jászai Mari tér 5.
Telefon: 530-800, 317-588

datacoop
SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET
2042 Diósd, Szabadság u. 11.
Telefon: 453–951
budapesti iroda: XII., Derkovits u. 3.
Telefon: 751–046, 751–019

EMO
ELEKTROMODUL
ELEKTROMODUL VÁLLALAT
ELEKTRONIKAI KERESKEDŐHÁZ
H–1390 BUDAPEST P.O.B. 158.
Telefon: 495–340, 497–730
Telex: 22-5154